

(19)



(11)

EP 3 764 226 B1

(12)

EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:

25.09.2024 Bulletin 2024/39

(21) Application number: **18914676.4**

(22) Date of filing: **12.04.2018**

(51) International Patent Classification (IPC):

G06F 9/455 ^(2018.01)	G06F 11/07 ^(2006.01)
G06F 11/14 ^(2006.01)	H04L 12/46 ^(2006.01)
H04L 43/00 ^(2022.01)	H04L 43/0805 ^(2022.01)
H04L 43/0817 ^(2022.01)	H04L 43/10 ^(2022.01)
H04L 43/20 ^(2022.01)	H04L 45/00 ^(2022.01)
H04L 45/50 ^(2022.01)	H04L 45/64 ^(2022.01)

(52) Cooperative Patent Classification (CPC):

G06F 9/45558; G06F 9/455; G06F 11/0712; G06F 11/0757; G06F 11/0793; G06F 11/142; G06F 11/1484; H04L 12/4633; H04L 12/4641; H04L 43/0805; H04L 43/10; H04L 43/14; H04L 43/20; H04L 45/64; G06F 2009/4557; (Cont.)

(86) International application number:

PCT/CN2018/082852

(87) International publication number:

WO 2019/196062 (17.10.2019 Gazette 2019/42)

(54) METHOD AND DEVICE FOR DETECTING STATE OF VIRTUAL MACHINE

VERFAHREN UND VORRICHTUNG ZUR ERKENNUNG DES ZUSTANDES EINER VIRTUELLEN MASCHINE

PROCÉDÉ ET DISPOSITIF DE DÉTECTION D'ÉTAT DE MACHINE VIRTUELLE

(84) Designated Contracting States:

AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR

(43) Date of publication of application:

13.01.2021 Bulletin 2021/02

(73) Proprietor: **Huawei Cloud Computing**

Technologies Co., Ltd.
Guizhou 550025 (CN)

(72) Inventor: **LI, Xiaowen**

Shenzhen, Guangdong 518129 (CN)

(74) Representative: **Maiwald GmbH**

Elisenhof
Elisenstraße 3
80335 München (DE)

(56) References cited:

WO-A1-2013/171552 CN-A- 101 383 688

CN-A- 105 357 038 CN-A- 106 980 528
US-A1- 2017 168 903 US-A1- 2017 257 266

- **MARTIN KLEPAC ET AL: "Enhancing Availability of Services Using Software-Defined Networking", ADVANCES IN ELECTRICAL AND ELECTRONIC ENGINEERING, vol. 13, no. 5, 23 December 2015 (2015-12-23), XP055769487, ISSN: 1336-1376, DOI: 10.15598/aeer.v13i5.1498**
- **CZIVA RICHARD ET AL: "SDN-Based Virtual Machine Management for Cloud Data Centers", IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE, US, vol. 13, no. 2, 1 June 2016 (2016-06-01), pages 212 - 225, XP011613000, ISSN: 1932-4537, [retrieved on 20160608], DOI: 10.1109/TNSM.2016.2528220**

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 3 764 226 B1

- **W DAVID ASHLEY: "Version 1.1 Libvirt Application Development Guide Using Python", 22 August 2016 (2016-08-22), XP055769311, Retrieved from the Internet
<URL:https://libvirt.org/docs/libvirt-appdev-guide-python/en-US/pdf/Version-1.1-Libvirt_Application_Development_Guide_Using_Python-en-US.pdf> [retrieved on 20210127]**

(52) Cooperative Patent Classification (CPC): (Cont.)
G06F 2009/45591; G06F 2009/45595;
H04L 43/0817; H04L 45/50; H04L 45/70

Description

TECHNICAL FIELD

[0001] This application relates to the field of virtual machines, and more specifically, to a virtual machine status probe method, and a device.

BACKGROUND

[0002] In a high availability (HA) virtual machine system, when resources in the virtual machine system are permitted, an HA process is allowed to be triggered to remotely recreate a virtual machine when the virtual machine in a source device is faulty, to further restore a corresponding service. However, in the virtual machine HA process, that the virtual machine in the source device is faulty should be ensured. Otherwise, if the virtual machine is incorrectly considered faulty but actually the virtual machine runs normally, after the virtual machine is recreated remotely, two virtual machines that use a same IP address and operate on a same volume exist in the system. To be specific, the virtual machines are called split-brain virtual machines. Because the split-brain virtual machines simultaneously operate on a same storage resource, unexpected abnormality may occur. In the prior art, two manners are usually used to determine whether a virtual machine in a source device is faulty, probe whether a network from a management node to a server in which the virtual machine is located is normal, or use a shared storage for a virtual machine in a cluster, and probe, in a storage plane, whether the virtual machine is faulty. However, incorrect determining or a failure may occur in both the manners.

[0003] Therefore, how to improve reliability of probing whether a virtual machine in the source device is faulty is an urgent problem to be resolved.

[0004] US 2017/168903 A1 relates to a method for live synchronization and management of virtual machines across computing and virtualization platforms to support disaster recovery. An "incremental forever" approach is combined with deduplication and synthetic full backups to speed up data transfer and update the disaster recovery sites.

[0005] US 2017/257266 A1 relates a method for sharing a JAVA virtual machine having multiple tenants. The method comprises allocating an IP address to each of the multiple tenants, respectively and creating a separate virtual network interface for each of the multiple tenants and configuring the allocated IP address. Respectively, for each of the multiple tenants, its network operation is performed via the virtual network interface created for that tenant.

SUMMARY

[0006] This application provides a virtual machine status probe method and a device, to improve reliability of

probing whether a virtual machine in the source device is faulty, thereby effectively avoiding occurrence of a virtual machine split brain.

[0007] The subject-matter for which protection is sought is defined by the appended claims. According to a first aspect, a virtual machine status probe method is provided. The method is used to: when a high availability procedure of a first virtual machine is triggered, determine whether to recreate the first virtual machine in a destination device, and the method is performed by the destination device, and includes:

obtaining an invocation message, where the invocation message is used to probe the first virtual machine, and obtaining an identifier of the first virtual machine; and
invoking a probe script based on the invocation message, and transferring the identifier of the first virtual machine to the probe script; sending a probe packet by running the probe script, where the probe packet is used to probe a status of the first virtual machine in the source device, and the probe packet includes the identifier of the first virtual machine; receiving feedback information sent by the source device, where the feedback information is used to indicate the status of the first virtual machine; and determining, based on the feedback information, whether to recreate the first virtual machine in the destination device.

[0008] Therefore, if a virtual machine high availability procedure is triggered, the destination device in which the first virtual machine needs to be recreated sends a probe packet to the source device in which the first virtual machine is located, where the probe packet is used to probe whether the first virtual machine runs in the source device. The destination device determines, based on the feedback information sent by the source device, whether to configure the first virtual machine in the destination device. This improves reliability of probing whether the virtual machine in the source device is faulty, and effectively avoids occurrence of a virtual machine split brain.

[0009] With reference to the first aspect, in some implementations of the first aspect, when a probe port of the source device and a probe port of the destination device are located in a same virtual local area network, the probe packet is a broadcast packet, and the probe packet includes a probe virtual local area network VLAN identifier of a service plane in which the first virtual machine is located.

[0010] With reference to the first aspect, in some implementations of the first aspect, the method further includes: receiving the probe virtual local area network VLAN identifier that is of the first service plane in which the first virtual machine is located and that is sent by a management device, configuring the probe port in the first service plane, and configuring the probe virtual local area network VLAN identifier for the probe port; and

the sending a probe packet by running the probe script includes: sending the probe packet through the probe port.

[0011] With reference to the first aspect, in some implementations of the first aspect, when the source device communicates with the destination device through an overlay network, the probe packet is a unicast packet, and the probe packet includes an identifier of a tunnel port of the source device.

[0012] With reference to the first aspect, in some implementations of the first aspect, the method further includes: obtaining, from the management device based on the identifier of the first virtual machine, the identifier of the tunnel port corresponding to the source device; and the sending a probe packet by running the probe script includes: sending the probe packet through a tunnel port of the destination device.

[0013] With reference to the first aspect, in some implementations of the first aspect, before the obtaining an invocation message, the method further includes: sending a first message to a server, where the first message is used to request disk storage information of the first virtual machine in the server, and is used to determine a status of the first virtual machine by probing a storage plane of the first virtual machine; and receiving no feedback information from the server within a predetermined time period.

[0014] In this case, when a high availability procedure of the virtual machine is triggered, if the system in which the virtual machine is located supports storage plane probe, the storage plane probe may be performed first. If the storage plane probe fails, the destination device in which the first virtual machine needs to be recreated may further send a probe packet to the source device in which the first virtual machine is located, to probe whether the first virtual machine runs normally in the source device. Therefore, if incorrect determining or a failure occurs when whether the virtual machine is faulty is probed in the storage plane, the destination device further sends the probe packet to the source device in which the first virtual machine is located, to probe whether the first virtual machine runs normally in the source device. This improves reliability of probing whether the virtual machine in the source device is faulty, and effectively avoids occurrence of a virtual machine split brain.

[0015] With reference to the first aspect, in some implementations of the first aspect, the determining, based on the feedback information, whether to recreate the first virtual machine in the destination device includes:

determining, when the feedback information indicates that the status of the first virtual machine in the source device is normal, not to recreate the first virtual machine in the destination device; or
determining, when the feedback information indicates that the status of the first virtual machine in the source device is faulty, to recreate the first virtual machine in the destination device.

[0016] According to a second aspect, a virtual machine status probe method is provided. The method is used to: when a high availability procedure of a first virtual machine is triggered, determine whether to recreate the first virtual machine in the destination device, and the method is performed by a source device, and includes: receiving a probe packet sent by the destination device, where the probe packet is used to probe a status of the first virtual machine in the source device, and the probe packet includes an identifier of the first virtual machine; querying for the status of the first virtual machine based on the probe packet; and sending feedback information to the destination device, where the feedback information is used to indicate the status of the first virtual machine in the source device.

[0017] With reference to the second aspect, in some implementations of the second aspect, when a probe port of the source device and a probe port of the destination device are located in a same virtual local area network, the probe packet is a broadcast packet, and the probe packet includes a probe virtual local area network VLAN identifier of a service plane in which the first virtual machine is located.

[0018] With reference to the second aspect, in some implementations of the second aspect, the method further includes: receiving the probe virtual local area network VLAN identifier that is of the first service plane in which the first virtual machine is located and that is sent by a management device, configuring the probe port in the first service plane, and configuring the probe virtual local area network VLAN identifier for the probe port; and the receiving a probe packet sent by the destination device includes: receiving, through the probe port, the probe packet sent by the destination device.

[0019] With reference to the second aspect, in some implementations of the second aspect, when the source device communicates with the destination device through an overlay network, the probe packet is a unicast packet, and the probe packet includes an identifier of a tunnel port of the source device.

[0020] With reference to the second aspect, in some implementations of the second aspect, the receiving a probe packet sent by the destination device includes: receiving, through the probe port, the probe packet sent by the destination device.

[0021] According to a third aspect, a system is provided. The system includes a destination device and a source device, and includes:

the destination device obtains invocation message when a high availability procedure of a first virtual machine is triggered, where the invocation message is used to probe the first virtual machine;
the destination device obtains an identifier of the first virtual machine;
the destination device invokes a probe script based on the invocation message, and transfers the identifier of the first virtual machine to the probe script;

the destination device sends a probe packet by running the probe script, where the probe packet is used to probe a status of the first virtual machine in the source device, and the probe packet includes the identifier of the first virtual machine;
 the source device receives the probe packet sent by the destination device;
 the source device queries for the status of the first virtual machine based on the probe packet;
 the source device sends feedback information to the destination device, where the feedback information is used to indicate the status of the first virtual machine in the source device;
 the destination device receives the feedback information sent by the source device; and
 the destination device determines, based on the feedback information, whether to recreate the first virtual machine in the destination device.

[0022] Therefore, if a virtual machine high availability procedure is triggered, the destination device in which the first virtual machine needs to be recreated sends a probe packet to the source device in which the first virtual machine is located, where the probe packet is used to probe whether the first virtual machine runs in the source device. The destination device determines, based on the feedback information sent by the source device, whether to configure the first virtual machine in the destination device. This improves reliability of probing whether the virtual machine in the source device is faulty, and effectively avoids occurrence of a virtual machine split brain.

[0023] With reference to the third aspect, in some implementations of the third aspect, when a probe port of the source device and a probe port of the destination device are located in a same virtual local area network, the probe packet is a broadcast packet, and the probe packet includes a probe virtual local area network VLAN identifier of a first service plane in which the first virtual machine is located.

[0024] With reference to the third aspect, in some implementations of the third aspect, the system further includes a management device. The management device receives the probe virtual local area network VLAN identifier of the first service plane, and sends the probe virtual local area network VLAN identifier to the source device and the destination device.

[0025] With reference to the third aspect, in some implementations of the third aspect, the destination device receives the probe virtual local area network VLAN identifier; the destination device configures the probe port in the first service plane, and configures the probe virtual local area network VLAN identifier for the probe port; and that the destination device sends a probe packet by running the probe script includes: the destination device sends the probe packet through the probe port.

[0026] With reference to the third aspect, in some implementations of the third aspect, the source device receives the probe virtual local area network VLAN identifier

sent by the management device; the source device configures the probe port in the first service plane, and configures the probe virtual local area network VLAN identifier for the probe port; and that the source device receives the probe packet sent by the destination device includes: the source device receives, through the probe port, the probe packet sent by the destination device.

[0027] With reference to the third aspect, in some implementations of the third aspect, the system further includes a switch; the controller sends the probe virtual local area network VLAN identifier that is of the first service plane in which the first virtual machine is located to the switch; the switch receives the probe virtual local area network VLAN identifier that is of the first service plane in which the first virtual machine is located and that is sent by the controller; the switch associates the VLAN identifier with an overlay network identifier; and the switch converts, based on the associated VLAN identifier and overlay network identifier, a packet that is received by the switch and that includes the VLAN identifier into a packet in an overlay form for sending.

[0028] With reference to the third aspect, in some implementations of the third aspect, when the source device communicates with the destination device through an overlay network, the probe packet is a unicast packet, and the probe packet includes an identifier of a tunnel port of the source device.

[0029] With reference to the third aspect, in some implementations of the third aspect, the system further includes a management device; the destination device obtains, from the management device based on the identifier of the first virtual machine, the identifier of the tunnel port corresponding to the source device; and that the destination device sends a probe packet by running the probe script includes: the destination device sends the probe packet through a tunnel port of the destination device.

[0030] With reference to the third aspect, in some implementations of the third aspect, that the source device receives the probe packet sent by the destination device includes: the source device receives the probe packet through the tunnel port of the source device.

[0031] With reference to the third aspect, in some implementations of the third aspect, before that the destination device invokes the invocation message, the method further includes: the destination device sends a first message to a server, where the first message is used to request disk storage information of the first virtual machine in the server, and is used to determine a status of the first virtual machine by probing a storage plane of the first virtual machine; and the destination device receives no feedback information from the server within a predetermined time period.

[0032] With reference to the third aspect, in some implementations of the third aspect, that the destination device determines, based on the feedback information, whether to recreate the first virtual machine in the destination device includes: the destination device deter-

mines, when the feedback information indicates that the status of the first virtual machine in the source device is normal, not to recreate the first virtual machine in the destination device; or the destination device determines, when the feedback information indicates that the status of the first virtual machine in the source device is faulty, to recreate the first virtual machine in the destination device.

[0033] According to a fourth aspect, a virtual machine fault probe method is provided. The method is used to: when a high availability procedure of a first virtual machine is triggered, determine whether to recreate the first virtual machine in a destination device, and the method is performed by a switch, and includes:

receiving a probe virtual local area network VLAN identifier that is of a first service plane in which the first virtual machine is located and that is sent by a controller;
 associating the VLAN identifier with an overlay network identifier; and
 converting, based on the associated VLAN identifier and overlay network identifier, a packet that is received by the switch and that includes the VLAN identifier into a packet in an overlay form for sending.

[0034] According to a fifth aspect, a destination device is provided. The destination device includes: an obtaining module, configured to obtain an invocation message, where the invocation message is used to probe the first virtual machine, where the obtaining module is further configured to obtain an identifier of the first virtual machine; an invocation module, configured to invoke a probe script based on the invocation message, and transfer the identifier of the first virtual machine to the probe script; a sending module, configured to send a probe packet by running the probe script, where the probe packet is used to probe a status of the first virtual machine in the source device, and the probe packet includes the identifier of the first virtual machine; a receiving module, configured to receive feedback information sent by the source device, where the feedback information is used to indicate the status of the first virtual machine; and a determining module, configured to determine, based on the feedback information, whether to recreate the first virtual machine in the destination device.

[0035] With reference to the fifth aspect, in some implementations of the fifth aspect, when a probe port of the source device and a probe port of the destination device are located in a same virtual local area network, the probe packet is a broadcast packet, and the probe packet includes a probe virtual local area network VLAN identifier of a service plane in which the first virtual machine is located.

[0036] With reference to the fifth aspect, in some implementations of the fifth aspect, the receiving module is further configured to receive the probe virtual local area network VLAN identifier that is of the first service plane

in which the first virtual machine is located and that is sent by a management device; and the destination device further includes a configuration module, configured to: configure the probe port in the first service plane, and configure the probe virtual local area network VLAN identifier for the probe port; and the sending module is specifically configured to: send the probe packet through the probe port.

[0037] With reference to the fifth aspect, in some implementations of the fifth aspect, when the source device communicates with the destination device through an overlay network, the probe packet is a unicast packet, and the probe packet includes an identifier of a tunnel port of the source device.

[0038] With reference to the fifth aspect, in some implementations of the fifth aspect, the obtaining module is further configured to obtain, from the management device based on the identifier of the first virtual machine, the identifier of the tunnel port corresponding to the source device; and the sending module is specifically configured to: send the probe packet through a tunnel port of the destination device.

[0039] With reference to the fifth aspect, in some implementations of the fifth aspect, the sending module is further configured to send a first message to a server, where the first message is used to request disk storage information of the first virtual machine in the server, and is used to determine a status of the first virtual machine by probing a storage plane of the first virtual machine; and the receiving module is further configured to receive no feedback information from the server within a predetermined time period.

[0040] With reference to the fifth aspect, in some implementations of the fifth aspect, the determining module is specifically configured to: determine, when the feedback information indicates that the status of the first virtual machine in the source device is normal, not to recreate the first virtual machine in the destination device; or determine, when the feedback information indicates that the status of the first virtual machine in the source device is faulty, to recreate the first virtual machine in the destination device.

[0041] According to a sixth aspect, a source device is provided. The source device includes: a receiving module, configured to receive a probe packet sent by a destination device, where the probe packet is used to probe a status of the first virtual machine in the source device, and the probe packet includes an identifier of the first virtual machine; a query module, configured to query for the status of the first virtual machine based on the probe packet; and a sending module, configured to send feedback information to the destination device, where the feedback information is used to indicate the status of the first virtual machine in the source device.

[0042] With reference to the sixth aspect, in some implementations of the sixth aspect, when a probe port of the source device and a probe port of the destination device

are located in a same virtual local area network, the probe packet is a broadcast packet, and the probe packet includes a probe virtual local area network VLAN identifier of a service plane in which the first virtual machine is located.

[0043] With reference to the six aspect, in some implementations of the six aspect, the receiving module is further configured to receive the probe virtual local area network VLAN identifier that is of the first service plane in which the first virtual machine is located and that is sent by a management device; and the source device further includes a configuration module, configured to: configure the probe port in the first service plane, and configure the probe virtual local area network VLAN identifier for the probe port; and

the receiving module is specifically configured to: receive, through the probe port, the probe packet sent by the destination device.

[0044] With reference to the six aspect, in some implementations of the six aspect, when the source device communicates with the destination device through an overlay network, the probe packet is a unicast packet, and the probe packet includes an identifier of a tunnel port of the source device.

[0045] With reference to the six aspect, in some implementations of the six aspect, the receiving module is specifically configured to: receive, through the tunnel port of the source device, the probe packet.

[0046] According to a seventh aspect, a destination device is provided. The destination device includes at least one processor, a memory, and an interface, where the interface is used for communication with a source device, the memory is configured to store computer program code, the computer program code includes an instruction, and the stored instruction is directly or indirectly executed by the at least one processor, so that the destination device can perform the method in any one of the first aspect or the optional implementations of the first aspect.

[0047] According to an eighth aspect, a source device is provided. The source device includes at least one processor, a memory, and an interface, where the interface is used for communication with a destination device, the memory is configured to store computer program code, the computer program code includes an instruction, and when the at least one processor executes the instruction, the source device performs the method in any one of the second aspect or the optional implementations of the second aspect.

[0048] According to a ninth aspect, a chip system is provided. The chip system includes at least one processor. The at least one processor is configured to execute a stored instruction, so that a destination device can perform the method in any one of the first aspect or the optional implementations of the first aspect.

[0049] According to a tenth aspect, a chip system is provided. The chip system includes at least one processor. The at least one processor is configured to execute

a stored instruction, so that a source device can perform the method in any one of the second aspect or the optional implementations of the second aspect.

[0050] According to an eleventh aspect, a computer program product is provided. The computer program product includes an instruction, and when the instruction is executed, a destination device can be enabled to perform the method in any one of the first aspect or the optional implementations of the first aspect.

[0051] According to a twelfth aspect, a computer program product is provided. The computer program product includes an instruction, and when the instruction is executed, a source device can be enabled to perform the method in any one of the second aspect or the optional implementations of the second aspect.

[0052] According to a thirteenth aspect, a computer storage medium is provided. The computer storage medium stores a program instruction, and when the instruction is executed, the destination device can perform the method in any one of the first aspect or the optional implementations of the first aspect.

[0053] According to a fourteenth aspect, a computer storage medium is provided. The computer storage medium stores a program instruction, and when the instruction is executed, the source device can perform the method in any one of the second aspect or the optional implementations of the second aspect.

BRIEF DESCRIPTION OF DRAWINGS

[0054]

FIG. 1 is a schematic block diagram of an architecture of a system for a virtual machine status probe method and a device according to this application;
 FIG. 2 is a schematic block diagram of a physical host according to this application;
 FIG. 3 is a schematic flowchart of a virtual machine status probe method according to this application;
 FIG. 4 is a schematic block diagram of an application scenario of a virtual machine status probe method according to this application;
 FIG. 5 is a schematic block diagram of an application scenario of a virtual machine status probe method according to this application;
 FIG. 6 is a schematic block diagram of an application scenario of a virtual machine status probe method according to this application;
 FIG. 7 is a schematic block diagram of an application scenario of a virtual machine status probe method according to this application;
 FIG. 8 shows a format of a VLAN packet according to this application;
 FIG. 9 shows a format of a VxLAN packet according to this application;
 FIG. 10 is a schematic block diagram of a destination device according to this application;
 FIG. 11 is a schematic block diagram of a source

device according to this application; and

FIG. 12 is a schematic block diagram of a device according to this application.

DESCRIPTION OF EMBODIMENTS

[0055] The following describes technical solutions of this application with reference to the accompanying drawings.

[0056] The technical solutions of the embodiments of this application may be applied to various communications systems, such as a global system for mobile communications (GSM) system, a code division multiple access (CDMA) system, a wideband code division multiple access (WCDMA) system, a general packet radio service (GPRS) system, a long term evolution (LTE) system, an LTE frequency division duplex (FDD) system, an LTE time division duplex (TDD) system, a universal mobile telecommunications system (UMTS), a worldwide interoperability for microwave access (WiMAX) communications system, a 5th generation (5G) system, and a new radio (NR) system.

[0057] A destination device and a source device in the embodiments of this application may be referred to as user equipment, an access terminal, a subscriber unit, a subscriber station, a mobile station, a mobile console, a remote station, a remote terminal, a mobile device, a user terminal, a terminal, a wireless communications device, a user agent, or a user apparatus. The terminal device may further be a cellular phone, a cordless phone, a session initiation protocol (SIP) phone, a wireless local loop (WLL) station, a personal digital assistant (PDA), a handheld device having a wireless communication function, a computing device, another processing device connected to a wireless modem, a vehicle-mounted device, a wearable device, a terminal device in a future 5G network, a terminal device in a future evolved public land mobile network (PLMN), or the like. This is not limited in the embodiments of this application.

[0058] FIG. 1 is a schematic block diagram of an architecture of a system 100 for a virtual machine status probe method and a device according to this application. As shown in FIG. 1, the architecture of the system 100 includes a management device 110, a destination device 120, and a source device 130. Different virtual machines may be created in the destination device 120 and the source device 130. The management device 110 is configured to manage the destination device 120 and the source device 130, for example, send, to the destination device 120 and the source device 130, some configuration information used to create and configure a virtual machine, and the like.

[0059] It should be noted that, FIG. 1 is merely a diagram of an example architecture. In addition to function units shown in FIG. 1, a plurality of other physical devices may further be included in the architecture of the system. This is not limited in this embodiment of this application.

[0060] A virtual machine status probe method provided

in this application may be applied to a destination device and a source device. The destination device and the source device each include a hardware layer, an operating system layer running on the hardware layer, and an application layer running on the operating system layer. The hardware layer includes hardware such as a central processing unit (CPU), a memory management unit (MMU), and a memory (also referred to as a main memory). The operating system may be any one or more types of computer operating systems that process a service by using a process, for example, a Linux operating system, a Unix operating system, an Android operating system, an iOS operating system, or a windows operating system. The application layer includes applications such as a browser, a contact list, text processing software, and instant communication software.

[0061] In addition, aspects or features of this application may be implemented as a method, an apparatus, or a product that uses standard programming and/or an engineering technology. The term "product" used in this application covers a computer program that can be accessed from any computer-readable component, carrier or medium. For example, the computer-readable medium may include but is not limited to: a magnetic storage component (for example, a hard disk, a floppy disk, or a magnetic tape), an optical disc (for example, a compact disc (CD), a digital versatile disc (DVD), a smart card, and a flash memory component (for example, an erasable programmable read-only memory (EPROM), a card, a stick, or a key drive). In addition, various storage media described in this specification may indicate one or more devices and/or other machine-readable media that are configured to store information. The term "machine-readable media" may include but is not limited to various media that can store, contain, and/or carry an instruction and/or data.

[0062] FIG. 2 is a schematic block diagram of a physical host according to this application. An example in which the physical host is the destination device 120 is used for description. The physical host 120 includes a hardware layer 121, a virtual machine monitor (VMM) 122 running on the hardware layer 121, and one or more virtual machines (VM) 123 running on the VMM 122. The hardware layer 121 may include a processor 1211, a physical memory 1212, a hard disk 1213, and a network interface device 1214. There may be one or more processors 1211. The network interface device 1214 may be a network adapter or a network interface card, configured to connect to any connectable network such as the internet or an enterprise network. The VM 123 may include a virtual processor 1231, a virtual memory 1232, a virtual hard disk 1233, and a guest operating system (Guest OS) 1234, where the Guest OS 1234 is an operating system running on virtual devices such as the virtual processor 1231, the virtual memory 1232, and the virtual hard disk 1233.

[0063] It should be understood that the physical host shown in FIG. 2 is merely an example, and should not

constitute a special limitation on the embodiments of this application. The physical host shown in FIG. 2 may be the destination device or the source device in this application. The destination device may also be referred to as a destination device, and the source device may also be referred to as a source device.

[0064] The following describes in detail a virtual machine status probe method provided in this application with reference to FIG. 3. FIG. 3 is a schematic flowchart of a virtual machine status probe method 200 according to an embodiment of this application. The method 200 may be applied to the system shown in FIG. 1, or certainly may be applied to another system. This is not limited in this embodiment of this application.

[0065] As shown in FIG. 3, the method 200 includes the following content.

[0066] In 210, when a high availability procedure of a first virtual machine is triggered, the destination device obtains invocation message, where the invocation message is used to probe the first virtual machine.

[0067] In 220, the destination device obtains an identifier of the first virtual machine.

[0068] In 230, the destination device invokes a probe script based on the invocation message, and transfers the identifier of the first virtual machine to the probe script.

[0069] In 240, the destination device sends a probe packet by running the probe script, where the probe packet is used to probe a status of the first virtual machine in the source device, and the probe packet includes the identifier of the first virtual machine.

[0070] In 250, the source device receives the probe packet sent by the destination device.

[0071] In 260, the source device queries for the status of the first virtual machine based on the probe packet.

[0072] In 270, the source device sends feedback information to the destination device, where the feedback information is used to indicate the status of the first virtual machine in the source device.

[0073] In 280, the destination device receives the feedback information sent by the source device.

[0074] In 290, the destination device determines, based on the feedback information, whether to recreate the first virtual machine in the destination device.

[0075] Therefore, in this embodiment of this application, if in an HA system, a virtual machine creation procedure is triggered in a destination device, the destination device in which a first virtual machine needs to be recreated sends a probe packet to a source device in which the first virtual machine is located, where the probe packet is used to probe whether the first virtual machine runs in the source device. The destination device determines, based on the feedback information sent by the source device, whether to configure the first virtual machine in the destination device. This improves reliability of probing whether the virtual machine in the source device is faulty, and effectively avoids occurrence of a virtual machine split brain.

[0076] It should be understood that the probing a virtual

machine status means probing a running status of the virtual machine in the source device, that is, whether the virtual machine is faulty and cannot run normally in the source device.

[0077] Optionally, when a probe port of the source device and a probe port of the destination device are located in a same virtual local area network, the probe packet is a broadcast packet, and the probe packet includes a probe virtual local area network (VLAN) identifier of a first service plane in which the first virtual machine is located.

[0078] Optionally, the destination device receives the probe virtual local area network VLAN identifier. The destination device configures the probe port in the first service plane, and configures the probe virtual local area network VLAN identifier for the probe port. That the destination device sends a probe packet by running the probe script includes: the destination device sends the probe packet through the probe port.

[0079] Optionally, the source device receives the probe virtual local area network VLAN identifier sent by a management device. The source device configures the probe port in the first service plane, and configures the probe virtual local area network VLAN identifier for the probe port. That the source device receives the probe packet sent by the destination device includes: the source device receives, through the probe port, the probe packet sent by the destination device.

[0080] Optionally, when the source device communicates with the destination device through an overlay network, the probe packet is a unicast packet, and the probe packet includes an identifier of a tunnel port of the source device.

[0081] Optionally, the system further includes a management device. The destination device obtains, from the management device based on the identifier of the first virtual machine, the identifier of the tunnel port corresponding to the source device. That the destination device sends a probe packet by running the probe script includes: the destination device sends the probe packet through a tunnel port of the destination device.

[0082] Optionally, that the source device receives the probe packet sent by the destination device includes: the source device receives the probe packet through the tunnel port of the source device.

[0083] Optionally, before that the destination device sends an invocation message, the method further includes: the destination device sends a first message to a server, where the first message is used to request disk storage information of the first virtual machine in the server, and is used to determine a status of the first virtual machine by probing a storage plane of the first virtual machine; and the destination device receives no feedback information from the server within a predetermined time period.

[0084] In this case, when a high availability procedure of the virtual machine is triggered, if the system in which the virtual machine is located supports storage plane probe, the storage plane probe may be performed first.

If the storage plane probe fails, the destination device in which the first virtual machine needs to be recreated may further send a probe packet to the source device in which the first virtual machine is located, to probe whether the first virtual machine runs normally in the source device. Therefore, if incorrect determining or a failure occurs when whether the virtual machine is faulty is probed in the storage plane, the destination device further sends the probe packet to the source device in which the first virtual machine is located, to probe whether the first virtual machine runs normally in the source device. This improves reliability of probing whether the virtual machine in the source device is faulty, and effectively avoids occurrence of a virtual machine split brain.

[0085] Optionally, that the destination device determines, based on the feedback information, whether to recreate the first virtual machine in the destination device includes: when the feedback information indicates that the status of the first virtual machine in the source device is normal, the destination device determines not to recreate the first virtual machine in the destination device; or when the feedback information indicates that the status of the first virtual machine in the source device is faulty, the destination device determines to recreate the first virtual machine in the destination device.

[0086] For clearer understanding of this application, the following describes the method 200 in detail in different application scenarios, for example, a VLAN scenario, an overlay scenario, and a scenario of interconnection with a third-party software-defined network (SDN) controller.

[0087] FIG. 4 is a schematic block diagram of an application scenario 300 of a virtual machine status probe method according to this application. The application scenario 300 is a virtual local area network (VLAN). A VLAN is a logical network segment divided from a local area network. A VLAN field (VLAN ID, VID) is used to identify a VLAN. A length of the VID is 12 bits and a value range is from 0 to 4095. 0 and 4095 are reserved values and cannot be used by a user. Therefore, one Ethernet can be divided into a maximum of 4094 VLANs. The application scenario 300 includes a management interface 310, a management node 320, a source device 330, and a destination device 340.

[0088] The management interface 310 is an interface-based installation and deployment tool, and an administrator uses the management interface 310 to configure a physical plane of a virtual machine and a probe identifier (where the probe identifier may also be referred to as a probe VLAN) of the physical plane.

[0089] The management node 320 includes a management configuration service module. The module is mainly configured to: receive configuration information that is delivered by the administrator by using the management interface 310, and trigger, based on the configuration information, a client in a corresponding host to implement configuration or modify a corresponding configuration. The management node 320 is responsible for

monitoring a running status of a virtual machine in each host, and when the management node 320 finds that a virtual machine in a host is faulty, an HA procedure may be triggered to re-configure the virtual machine in another host.

[0090] A first virtual machine (Virtual Machine 1, VM 1) is configured in the source device 330, and the source device further includes a probe module 331, a client 332, and a probe port 333. The probe module 331 may provide a capability of sending a user-defined broadcast probe packet, and receive a probe packet to probe whether a virtual machine is alive. The client 332 is configured to: receive the configuration information sent by the management node 320, and perform a related operation to complete a configuration change. The probe port 333 is a probe port of a corresponding physical plane, and the probe module 331 sends and receives a packet through the probe port 333.

[0091] The destination device 340 includes a probe module 341, a client 342, a probe port 343, and a management module 344. For the probe module 341, the client 342, and the probe port 343, refer to corresponding descriptions of those in the source device 330. Details are not described herein again. The management module 344 is responsible for computing virtualization management, and is configured to create and manage a virtual machine deployed in the destination device.

[0092] It should be understood that the application scenario shown in FIG. 4 is merely for clearer understanding of this application, and should not constitute a special limitation on this embodiment of this application. For example, in addition to the source device 330 and the destination device 340, another host may further be managed by the management node, and the application scenario 300 may further include another management node. Other virtual machines may be further configured in the source device 330 and the destination device 340. Modular units configured in the source device 330 and the destination device 340 should be consistent. The source device 330 may alternatively be a destination device, and the destination device 340 may alternatively be a source device. For example, if a VM 2 is configured in the destination device 340, and if the VM 2 needs to be configured in the source device 330 in an HA application scenario, the destination device 340 may be referred to as a source device, and the source device 330 may be referred to as a destination device.

[0093] For clearer understanding of this application, the following describes a specific procedure of this application in detail.

1. The administrator configures a probe identifier of a service plane physnet 2 by using the management interface 310 (where herein, it is assumed that a physical plane corresponding to the VM 1 is physnet 2; when creating a virtual machine, the user needs to specify a network VLAN used by the virtual machine to perform service communication; the net-

work is mapped to a corresponding physical plane; a mapping relationship between the network and the physical plane is stored on the management node 320; and the virtual machine finally performs communication by using the physical plane, and the physical plane is also referred to as a service plane.) and sends the probe identifier of the service plane physnet 2 to the management node 320.

2. The management node 320 receives the probe identifier of the service plane physnet 2 and sends the probe identifier of the service plane physnet 2 to each host managed by the management node, for example, the source device 330 and the destination device 340 shown in FIG. 4.

3. A client of each host creates a probe port corresponding to the probe identifier in the service plane physnet 2 and the probe port may use the probe identifier as a mark. The probe port 333 of the source device shown in FIG. 3 is a probe port of the service plane physnet 2, a mark of the probe port 333 is the probe identifier of the service plane physnet 2, the probe port 343 of the destination device is a probe port of the service plane physnet 2, and a mark of the probe port 343 is the probe identifier of the service plane physnet 2.

Optionally, each host may alternatively create, in the service plane physnet 2 by using a custom script or in another manner, a probe port corresponding to the probe identifier. A specific method for creating a probe port in the service plane by the host is not limited in this application.

4. A probe module on a virtual machine in a cluster monitors the probe port of the service plane physnet 2. This step is not shown in FIG. 3.

5. When a management configuration service module in the management node 320 detects that the VM 1 in the source device 330 is faulty, an HA procedure is triggered, and when the management configuration service module determines to configure the VM 1 in the destination device 340, the management module 344 in the destination device 340 sends an invocation message to the probe module 341, where the invocation message is used to trigger the probe module 341 to execute a probe script, and the invocation message includes an identifier of the VM 1 and an identifier of a service plane in which the VM 1 is located.

[0094] The identifier of the service plane in which the VM 1 is located is determined by obtaining, by the management module 344, a network used by the VM to perform communication and based on a mapping relationship between the network and a physical plane.

[0095] Optionally, the invocation message may be an application programming interface (API), and the application programming interface triggers the probe module 341 to execute the probe script.

[0096] Optionally, before the management module 344

in the destination device 340 sends the invocation message to the probe module 341, the destination device 340 triggers storage plane probe for the VM 1, but receives no feedback information from a storage server 350 within a specified time. Further, the management module 344 sends the invocation message to the probe module 341, and probes the virtual machine in the service plane.

[0097] The following describes, with reference to FIG. 5, how the destination device 340 performs storage plane probe. As shown in FIG. 5, the application scenario 300 may further include a server 350. The server 350 is a shared storage unit in the application scenario 300. Each virtual machine in the application scenario 300 uses shared storage. A storage space in the server 350 is allocated to each virtual machine, and when the virtual machine runs normally, the storage space corresponding to the virtual machine is locked. The storage space corresponding to the virtual machine is not locked only when the virtual machine is abnormal. For example, at a moment t_1 , the VM 1 is created in the source device 330, the VM 1 obtains a storage space 1 in the server 350, and the storage space 1 is locked at the moment t_1 . If the HA application scenario is caused by a non-virtual machine fault, when the destination device 340 determines to configure the VM 1, the destination device 340 obtains the storage space 1 from the server 350 when creating the VM 1, but the storage space 1 is locked. Therefore, the destination device 340 no longer creates the VM1. This can avoid occurrence of a virtual machine split brain.

[0098] However, the storage plane probe depends on the shared storage. If the virtual machine uses local storage, the storage plane probe cannot be performed. In addition, if the server that provides the shared storage is faulty, or a software application scenario of the shared storage is faulty, probe on the storage plane may be affected. Therefore, when the storage plane probe is abnormal, or the application scenario in which the virtual machine is located does not support the storage plane probe, a virtual machine running status may be probed by using a service plane in which the virtual machine is located, in other words, as described above, in the HA application scenario, when the destination device 340 determines to configure the VM 1, the management module 344 in the destination device 340 sends an invocation message to the probe module 341, where the invocation message is used to trigger the probe module 341 to execute a probe script, and the invocation message includes the identifier of the VM 1 and the identifier of the service plane in which the VM 1 is located. Therefore, if incorrect determining or a failure occurs when whether the virtual machine is faulty is probed in the storage plane, the destination device further sends a probe packet to the source device in which the first virtual machine is located, to probe whether the first virtual machine runs normally in the source device. This improves reliability of probing whether the virtual machine in the source de-

vice is faulty, and effectively avoids occurrence of a virtual machine split brain.

[0099] Optionally, the identifier of the VM 1 may be an IP address of the VM 1, or may be a label of the VM 1, or a type of mark information. In this embodiment of this application, an example in which the identifier of the VM 1 is the IP address of the VM 1 is used for description.

[0100] 6. The probe module 341 in the destination device 340 sends the probe packet to the probe port 343. The probe packet is a broadcast packet, and the probe packet includes the identifier of the VM 1 and the identifier of the service plane in which the VM 1 is located.

[0101] Optionally, a destination MAC address of the probe packet is "ff:ff:ff:ff:ff:ff", indicating that the probe packet is the broadcast packet.

[0102] 7. The probe packet is received through the probe port 333, of the service plane physnet 2, in the source device 330.

[0103] 8. After the probe packet sent by the destination device 340 is received through the probe port 333, of the service plane physnet 2, in the source device 330, the probe packet is sent to the probe module 331, and the probe module 331 obtains the identifier of the VM 1 carried in the probe packet, queries, by using a libvirt interface, whether the corresponding VM 1 runs normally, and feeds back a query result to the destination device 340. A transfer path of the feedback information is opposite to a transfer path of the probe packet. To be specific, the probe module 331 sends the feedback information to the probe port 333, the probe port 343 of the destination device 340 receives the feedback information, and sends the feedback information to the probe module 341, and the probe module 341 transfers the feedback information to the management module 344.

[0104] Optionally, a command for querying, by using the libvirt interface, whether the corresponding VM 1 runs normally may be:

```
(libvirt.open("qemu:///system").lookupByUUIDString(v  
m_uuid).isActive()).
```

[0105] When receiving the command, the VM 1 sends feedback information to the probe module 331 based on a running status of the VM 1, where the feedback information may be true or false, where true indicates that the virtual machine VM 1 runs normally, and false indicates that the virtual machine VM 1 is faulty.

[0106] 9. The management module 344 in the destination device 340 determines, based on whether the VM 1 runs normally that is fed back by the source device 330, whether to configure the virtual machine VM 1 in the destination device 340; and

when the feedback information indicates that the VM 1 runs in the source device 330, determines not to configure the VM 1 in the destination device 340; or when the feedback information indicates that the VM 1 does not run in the source device 330, determines to configure the VM 1 in the destination device 340.

[0107] Therefore, in this embodiment of this application, if in an HA application scenario, a virtual machine creation procedure is triggered in a destination device, the destination device in which a first virtual machine needs to be recreated sends a probe packet to a source device in which the first virtual machine is located, where the probe packet is used to probe whether the first virtual machine runs in the source device. The destination device determines, based on the feedback information sent by the source device, whether to configure the first virtual machine in the destination device. This improves reliability of probing whether the virtual machine in the source device is faulty, and effectively avoids occurrence of a virtual machine split brain.

[0108] FIG. 6 is schematic block diagram of an application scenario 400 of a virtual machine status probe method according to this application. The application scenario 400 is an overlay network. An existing overlay network mainly includes a virtual extensible local area network (virtual extensible LAN, VxLAN), network virtualization using generic routing encapsulation (NVGRE), and stateless transport tunneling (STT). The application scenario 400 includes a management node 410, a source device 420, and a destination device 430.

[0109] The management node 410 includes a network management service module, and the network management service module is mainly configured to create and manage a related network resource.

[0110] A first virtual machine (Virtual Machine 1, VM 1) is configured in the source device 420, and the source device further includes a probe module 421 and a tunnel 422 (where the tunnel 422 may also be referred to as a data network plane interface). The probe module 421 may provide a capability of sending a user-defined broadcast probe packet, and receive a probe packet to probe whether a virtual machine is alive. The tunnel 422 is configured to send and receive a packet.

[0111] The destination device 430 includes a probe module 431, a tunnel 432, and a management module 433. For the probe module 431 and the tunnel 432, refer to corresponding descriptions of those in the source device 420. Details are not described herein again. The management module 433 is responsible for computing virtualization management, and is configured to create and manage a virtual machine deployed in the destination device.

[0112] It should be understood that the application scenario shown in FIG. 6 is merely for clearer understanding of this application, and should not constitute a special limitation on this embodiment of this application. For example, in addition to the source device 420 and the destination device 430, another host may further be managed by the management node, and the application scenario 400 may further include another management node. Other virtual machines may be further configured in the source device 420 and the destination device 430. Modular units configured in the source device 420 and the destination device 430 should be consistent. The

source device 420 may alternatively be a destination device, and the destination device 430 may alternatively be a source device. For example, if a VM 2 is configured in the destination device 430, and if the VM 2 needs to be configured in the source device 420 in an HA application scenario, the destination device 430 may be referred to as a source device, and the source device 420 may be referred to as a destination device.

[0113] For clearer understanding of this application, the following describes a specific procedure of this application in detail.

1. In a virtual machine application scenario, each host starts a probe module, and the probe module in each host monitors a tunnel, to identify a probe packet.
2. When the VM 1 is determined to be configured in the destination device 430 in the HA application scenario, the management module 433 in the destination device 430 obtains, from the network management service module in the management node 410, an identifier of a tunnel in the source device 420 in which the VM 1 is located.

[0114] Optionally, the management module 433 in the destination device 430 may obtain, based on an API provided by the management node 410, the identifier of the tunnel in the source device 420 in which the VM1 is located.

[0115] Optionally, the identifier of the tunnel may be an IP address of the tunnel, or may be a label of the tunnel, or may be a type of mark information. In this embodiment of this application, an example in which the identifier of the tunnel is the IP address of the tunnel is used for description.

[0116] Optionally, before the management module 433 in the destination device 430 obtains, from the network management service module in the management node 410, the identifier of the tunnel in the source device 420 in which the VM 1 is located, the destination device 430 triggers storage plane probe for the VM 1, but receives no feedback information from a storage server within a specified time. Further, the management module 433 obtains, from the network management service module in the management node 410, the identifier of the tunnel in the source device 420 in which the VM 1 is located.

[0117] For a specific process in which the destination device 430 triggers storage plane probe for the VM 1, refer to related descriptions in FIG. 4. To avoid repetition, details are not described herein again.

[0118] 3. The management module 433 in the destination device 430 sends an invocation message to the probe module 431, where the invocation message is used to trigger the probe module 431 to send a probe packet, and the invocation message includes an identifier of the VM 1 and the identifier of the tunnel in the source device 420 in which the VM 1 is located.

[0119] Optionally, the invocation message may be an

application programming interface (API), and the application programming interface triggers the probe module 431 to execute the probe script.

[0120] Optionally, the identifier of the VM 1 may be an IP address of the VM 1, or may be a label of the VM 1, or a type of mark information. In this embodiment of this application, an example in which the identifier of the VM 1 is the IP address of the VM 1 is used for description.

[0121] 4. The probe module 431 in the destination device 430 sends the probe packet, where the probe packet is a unicast probe packet, a destination address of the probe packet is an IP address of the source device 420, and the probe packet further includes the identifier of the virtual machine.

[0122] 5. The probe packet is received through the tunnel 422 in the source device 420.

[0123] 6. The probe packet is sent to the probe module 421 through the tunnel 422 in the source device 420.

[0124] 7. The probe module 421 obtains the identifier of the VM 1 carried in the probe packet, queries, by using a libvirt interface, whether the corresponding VM 1 runs normally, and feeds back a query result to the destination device 430. A transfer path of the feedback information is opposite to a transfer path of the probe packet. To be specific, the probe module 421 sends the feedback information to the tunnel 422, the tunnel 432 of the destination device 430 receives the feedback information, and sends the feedback information to the probe module 431, and the probe module 431 transfers the feedback information to the management module 433.

[0125] Optionally, a command for querying, by using the libvirt interface, whether the corresponding VM 1 runs normally may be:

```
(libvirt.open("qemu:///system") .lookupByUUIDString(v
m_uuid).isActive()).
```

[0126] When receiving the command, the VM 1 sends feedback information to the probe module 421 based on a running status of the VM 1, where the feedback information may be true or false, where true indicates that the virtual machine VM 1 runs normally, and false indicates that the virtual machine VM 1 is faulty.

[0127] 8. The management module 433 in the destination device 430 determines, based on whether the VM 1 runs normally that is fed back by the source device 420, whether to configure the virtual machine VM 1 in the destination device 430; and

when the feedback information indicates that the VM 1 runs in the source device 420, determines not to configure the VM 1 in the destination device 430; or when the feedback information indicates that the VM 1 does not run in the source device 420, determines to configure the VM 1 in the destination device 430.

[0128] Therefore, in this embodiment of this application, if in an HA application scenario, a virtual machine creation procedure is triggered in a destination device, the destination device in which a first virtual machine

needs to be recreated sends a probe packet to a source device in which the first virtual machine is located, where the probe packet is used to probe whether the first virtual machine runs in the source device. The destination device determines, based on the feedback information sent by the source device, whether to configure the first virtual machine in the destination device. This improves reliability of probing whether the virtual machine in the source device is faulty, and effectively avoids occurrence of a virtual machine split brain.

[0129] FIG. 7 is schematic block diagram of an application scenario 500 of a virtual machine status probe method and a device according to this application. The application scenario 500 is a virtual extensible local area network (virtual extensible LAN, VxLAN) in virtual inter-connection with a third-party software-defined network (SDN). The application scenario 500 includes a management node 510, a configuration interface 520, a management interface 530, a source device 540, a destination device 550, a switch 560, a switch 570, and a third-party controller 580. In this application scenario, a VxLAN network is mapped as a VLAN network. It should be understood that the VxLAN network is only a form of an overlay network. In this application, another overlay may alternatively be mapped as the VLAN network. For a specific implementation of mapping the another overlay as the VLAN network, refer to a process of mapping the VxLAN network as the VLAN network. Details are not described herein again.

[0130] The management node 510 includes a network management service module 511 and a management configuration service module 512. The network management service module 511 is mainly configured to create and manage a related network resource. The management configuration service module 512 is mainly configured to receive configuration information delivered by an administrator by using the management interface 530, and trigger, based on the configuration information, a client in a corresponding host to implement configuration or modify a corresponding configuration. When a virtual machine is created, when an administrator specifies a service network of the virtual machine as a VxLAN network, the network management service module 511 maps the VxLAN network as a VLAN network through hierarchical binding, so that a packet of the virtual machine can be forwarded in a manner in the VLAN network, and when the packet is sent to a switch, the packet is forwarded in a manner of forwarding the packet in the VxLAN network.

[0131] For the configuration interface 520, the administrator may perform, by using the configuration interface 520, interconnection configuration between a third-party SDN controller and the VxLAN. The administrator sends, by using the configuration interface 520 to the network management service module 511 in the management node 510, an identifier of a service plane mapped from the VxLAN network. The network management service module 511 stores a mapping relationship of the service

plane mapped from the VxLAN network.

[0132] The management interface 530 is an interface-based installation and deployment tool. The administrator sends, by using the management interface to the destination device 550, the identifier of the service plane mapped from the VxLAN network. The management module 554 in the destination device 550 records the mapping relationship of the service plane mapped from the VxLAN network. The administrator uses the management interface 530 to configure a physical plane of a virtual machine and a probe identifier (where the probe identifier may also be referred to as a probe VLAN) of the physical plane.

[0133] A first virtual machine (Virtual Machine 1, VM 1) is configured in the source device 540, and the source device further includes a probe module 541, a client 542, and a probe port 543. The probe module 541 may provide a capability of sending a user-defined broadcast probe packet, and receive a probe packet to probe whether a virtual machine is alive. The client 542 is configured to receive the configuration information sent by the management node 510, and perform a related operation to complete a configuration change. The probe port 543 is a probe port of a corresponding physical plane, and the probe module 541 sends and receives a packet through the probe port 543.

[0134] The destination device 550 includes a probe module 551, a client 552, a probe port 553, and a management module 554. For the probe module 551, the client 552, and the probe port 553, refer to corresponding descriptions of those in the source device 540. Details are not described herein again. The management module 544 is responsible for computing virtualization management, and is configured to create and manage a virtual machine deployed in the destination device.

[0135] The switch 560 and the switch 570 are configured to transmit a packet.

[0136] The third-party controller 580 may implement configuration of the switch 560 and the switch 570.

[0137] It should be understood that the application scenario shown in FIG. 7 is merely for clearer understanding of this application, and should not constitute a special limitation on this embodiment of this application. For example, in addition to the source device 540 and the destination device 550, another host may further be managed by the management node, and the application scenario 500 may further include another management node. Other virtual machines may be further configured in the source device 540 and the destination device 550. Modular units configured in the source device 540 and the destination device 550 should be consistent. The source device 540 may alternatively be a destination device, and the destination device 550 may alternatively be a source device.

[0138] For clearer understanding of this application, the following describes a specific procedure of this application in detail.

1. The administrator may perform, by using the configuration interface 520, interconnection configuration between a third-party SDN controller and a VxLAN. The administrator sends, by using the configuration interface 520 to the network management service module 511 in the management node 510, an identifier of a service plane mapped from the VxLAN network (where herein, it is assumed that the service plane is a physnet 2). The network management service module 511 stores a mapping relationship of the service plane mapped from the VxLAN network.

2. The network management service module 511 in the management node 510 maps the VxLAN network as a VLAN network through hierarchical binding.

3. The administrator sends, by using the management interface 530 to the destination device 550, the identifier of the service plane mapped from the VxLAN network. The management module 554 in the destination device 550 records a mapping relationship of the service plane mapped from the VxLAN network.

4. The administrator configures a Layer 2 bridge (Lay2bridge, L2BR) (where the L2BR is configured to implement Layer 2 interoperation between servers of the VxLAN and the VLAN) in the SDN controller to connect a probe identifier to a VNI. The L2BR encapsulates the packet into a VxLAN packet based on VLAN-VNI mapping predefined by the controller, searches a vxlan flow table of the destination device, and forwards the packet to a destination virtual tunnel end point (Virtual Tunnel End Points, VTEP).

5. The administrator configures a probe identifier of the service plane physnet 2 by using the management interface 530, and sends the probe identifier of the service plane physnet 2 to the management node 510.

6. The management configuration service module 512 in the management node 510 receives the probe identifier of the service plane physnet 2 and sends the probe identifier of the service plane physnet 2 to each host managed by the management node, for example, the source device 540 and the destination device 550 shown in FIG. 5.

7. A client of each host creates a probe port corresponding to the probe identifier in the service plane physnet 2 and the probe port may use the probe identifier as a mark. The probe port 543 of the source device shown in FIG. 6 is a probe port of the service plane physnet 2, a mark of the probe port 543 is the probe identifier of the service plane physnet 2, the probe port 553 of the destination device is a probe port of the service plane physnet 2, and a mark of the probe port 553 is the probe identifier of the service plane physnet 2.

Optionally, each host may alternatively create, in the service plane physnet 2 by using a custom script or in another manner, a probe port corresponding to

the probe identifier. A specific method for creating a probe port in the service plane by the host is not limited in this application.

8. A probe module on a virtual machine in a cluster monitors the probe port of the service plane physnet 2. This step is not shown in FIG. 6.

9. When it is determined to configure the VM 1 in the destination device 550 in an HA application scenario, the management module 554 in the destination device 550 sends an invocation message to the probe module 551, where the invocation message is used to trigger the probe module 551 to execute a probe script, and the invocation message includes an identifier of the VM 1 and an identifier of a service plane in which the VM 1 is located.

[0139] Optionally, the invocation message may be an application programming interface (API), and the application programming interface triggers the probe module 551 to execute the probe script.

[0140] The identifier of the service plane in which the VM 1 is located is determined by obtaining, by the management module 454, a network used by the VM to perform communication and based on a mapping relationship between the network and a physical plane.

[0141] Optionally, before the management module 554 in the destination device 550 sends the invocation message to the probe module 551, the destination device 550 triggers storage plane probe for the VM 1, but receives no feedback information from a storage server within a specified time. Further, the management module 554 sends the invocation message to the probe module 551, and probes the virtual machine in the service plane.

[0142] For a specific procedure in which the destination device 550 triggers storage plane probe for the VM 1, refer to related descriptions in FIG. 4. Details are not described herein again.

[0143] 10. The probe module 551 in the destination device 550 sends the probe packet to the probe port 553. The probe packet is a broadcast packet, and the probe packet includes the identifier of the VM 1 and the identifier of the service plane in which the VM 1 is located.

[0144] 11. The broadcast packet sent by the probe port 553 is transferred to the switch 560, and the switch 560 identifies the probe packet based on a VLAN probe identifier carried in the probe packet, encapsulates the probe packet from an original VLAN network packet format into a VxLAN network packet format, and sends the encapsulated packet to the switch 570.

[0145] Specifically, the packet format of the VLAN probe packet is shown in FIG. 8. The packet format is defined in an 802.1Q standard, and the VLAN packet includes a destination MAC address, a source MAC address, an 802.1Q tag, a type, data, and a check bit FCS. An 802.1Q field is added to an Ethernet frame format defined in the 802.1Q standard based on a traditional Ethernet frame format. The field includes four parts: Type, PRI, CFI, and VID. Meanings of the four parts are

as follows:

[0146] Type has a length of two bytes, and indicates a frame type. A Type field in an 802.1Q tag frame has a fixed value of 0x8100. If a device that does not support 802.1Q receives an 802.1Q frame, the device discards the frame.

[0147] A priority (PRI) field has a length of three bits, indicates a priority of the Ethernet frame, and has a value ranging from 0 to 7. A larger value indicates a higher priority. When transmission congestion occurs in a switch or router, the switch or router preferentially sends a high-priority data frame.

[0148] A canonical format indicator (CFI) has a length of one bit, and indicates whether a MAC address is in a classic format. A CFI having a value of 0 indicates that a MAC address is in a classic format. A CFI having a value of 1 indicates that a MAC address is not in a classic format. This field is used to distinguish between an Ethernet frame, an FDDI frame, and a token ring network frame. In the Ethernet frame, a CFI has a value of 0.

[0149] A local area network ID (VLAN ID, VID) has a length of 12 bits, and has a value ranging from 0 to 4095. 0 and 4095 are reserved values and cannot be used. This field uniquely identifies a VLAN. The VID having a length of 12 bits can represent 4096 different values. One Ethernet can be divided into a maximum of 4094 VLANs other than VLANs of the two reserved values.

[0150] The VxLAN encapsulates, in a physical network, a data frame for communication in a logical network for transmission, and a process of encapsulation and decapsulation is completed by the switch 560. After adding a VxLAN header to the data frame in the logical network, the VxLAN encapsulates the data frame into a UDP packet in the physical network for transmission. A format of the VxLAN header is shown in FIG. 9. The VxLAN header includes eight bytes, and the first byte is a flag bit. If a flag bit I is set to 1, it indicates that the VxLAN header is valid. Other flags are reserved and should be set to 0 during transmission. The second byte to the fourth byte are reserved. The fifth byte to the seventh byte are VxLAN identifiers, used to indicate a unique logical network. The eighth byte is also a reserved field and is not used temporarily.

[0151] After adding the VxLAN header to the data frame of the logical link network, the switch 560 sequentially adds a user datagram protocol (UDP) header, an IP header, and an Ethernet frame header, and then sends the data frame to the switch 570.

[0152] It should be understood that if there are a plurality of hosts managed by the management node, and each host corresponds to a corresponding switch, the encapsulated packet is sent to the plurality of switches.

[0153] 12. The switch 560 sends the encapsulated packet to the switch 570.

[0154] 13. The switch 570 sends the probe packet to the source device 540.

[0155] 14. The probe port 543, of the service plane physnet 2, in the source device 540, receives the probe

packet.

[0156] 15. After receiving the probe packet sent by the destination device 550, the probe port 543, of the service plane physnet 2, in the source device 540, sends the probe packet to the probe module 541, and the probe module 541 obtains the identifier of the VM 1 carried in the probe packet, queries, by using a libvirt interface, whether the corresponding VM 1 runs normally, and feeds back a query result to the destination device 550. A transfer path of the feedback information is opposite to a transfer path of the probe packet. To be specific, the probe module 541 sends the feedback information to the probe port 543, then the switch 570, the switch 560, and the probe port 553 that is of the destination device 550 sequentially receive the feedback information, and the probe port 553 sends the feedback information to the probe module 551, and the probe module 551 transfers the feedback information to the management module 554.

[0157] Optionally, a command for querying, by using the libvirt interface, whether the corresponding VM 1 runs normally may be:

```
(libvirt.open("qemu:///system") .lookupByUUIDString(v
m_uuid).isActive()).
```

[0158] When receiving the command, the VM 1 sends feedback information to the probe module 541 based on a running status of the VM 1, where the feedback information may be true or false, where true indicates that the virtual machine VM 1 runs normally, and false indicates that the virtual machine VM 1 is faulty.

[0159] 16. The management module 554 in the destination device 550 determines, based on whether the VM 1 runs normally that is fed back by the source device 540, whether to configure the virtual machine VM 1 in the destination device 550.

[0160] Therefore, in this embodiment of this application, if in an HA application scenario, a virtual machine creation procedure is triggered in a destination device, the destination device in which a first virtual machine needs to be recreated sends a probe packet to a source device in which the first virtual machine is located, where the probe packet is used to probe whether the first virtual machine runs in the source device. The destination device determines, based on the feedback information sent by the source device, whether to configure the first virtual machine in the destination device. This improves reliability of probing whether the virtual machine in the source device is faulty, and effectively avoids occurrence of a virtual machine split brain.

[0161] This application further provides a system. The system includes a destination device and a source device, and includes:

[0162] The destination device obtains invocation message when a high availability procedure of a first virtual machine is triggered, where the invocation message is used to probe the first virtual machine. The destination device obtains an identifier of the first virtual machine. The destination device invokes a probe script based on

the invocation message, and transfers the identifier of the first virtual machine to the probe script. The destination device sends a probe packet by running the probe script, where the probe packet is used to probe a status of the first virtual machine in the source device, and the probe packet includes the identifier of the first virtual machine. The source device receives the probe packet sent by the destination device. The source device queries for the status of the first virtual machine based on the probe packet. The source device sends feedback information to the destination device, where the feedback information is used to indicate the status of the first virtual machine in the source device. The destination device receives the feedback information sent by the source device. The destination device determines, based on the feedback information, whether to recreate the first virtual machine in the destination device.

[0163] Optionally, when a probe port of the source device and a probe port of the destination device are located in a same virtual local area network, the probe packet is a broadcast packet, and the probe packet includes a probe virtual local area network identifier of a first service plane in which the first virtual machine is located.

[0164] Optionally, the system further includes a management device. The management device receives the probe virtual local area network VLAN identifier of the first service plane, and sends the probe virtual local area network VLAN identifier to the source device and the destination device.

[0165] Optionally, the destination device receives the probe virtual local area network VLAN identifier.

[0166] The destination device configures the probe port in the first service plane, and configures the probe virtual local area network VLAN identifier for the probe port. That the destination device sends a probe packet by running the probe script includes: the destination device sends the probe packet through the probe port.

[0167] Optionally, the source device receives the probe virtual local area network VLAN identifier sent by the management device. The source device configures the probe port in the first service plane, and configures the probe virtual local area network VLAN identifier for the probe port. That the source device receives the probe packet sent by the destination device includes: the source device receives, through the probe port, the probe packet sent by the destination device.

[0168] Optionally, the system further includes a switch. The controller sends the probe virtual local area network VLAN identifier of the first service plane in which the first virtual machine is located to the switch. The switch receives the probe virtual local area network VLAN identifier that is of the first service plane in which the first virtual machine is located and that is sent by the controller, associates the VLAN identifier with an overlay network identifier, and converts, based on the associated VLAN identifier and overlay network identifier, a packet that is received by the switch and that includes the VLAN identifier into a packet in an overlay form for sending.

[0169] Optionally, when the source device communicates with the destination device through an overlay network, the probe packet is a unicast packet, and the probe packet includes an identifier of a tunnel port of the source device.

[0170] Optionally, the system further includes a management device. The destination device obtains, from the management device based on the identifier of the first virtual machine, the identifier of the tunnel port corresponding to the source device. That the destination device sends a probe packet by running the probe script includes: the destination device sends the probe packet through a tunnel port of the destination device.

[0171] Optionally, that the source device receives the probe packet sent by the destination device includes: the source device receives the probe packet through the tunnel port of the source device.

[0172] Optionally, before that the destination device sends an invocation message, the method further includes: the destination device sends a first message to a server, where the first message is used to request disk storage information of the first virtual machine in the server, and is used to determine a status of the first virtual machine by probing a storage plane of the first virtual machine; and the destination device receives no feedback information from the server within a predetermined time period.

[0173] Optionally, that the destination device determines, based on the feedback information, whether to recreate the first virtual machine in the destination device includes: when the feedback information indicates that the status of the first virtual machine in the source device is normal, the destination device determines not to recreate the first virtual machine in the destination device; or when the feedback information indicates that the status of the first virtual machine in the source device is faulty, the destination device determines to recreate the first virtual machine in the destination device.

[0174] It should be understood that, for understanding of the source device and the destination device in the system, refer to the descriptions in the foregoing embodiments. To avoid repetition, details are not described herein again.

[0175] FIG. 10 is a schematic block diagram of a destination device 600 according to this application. As shown in FIG. 10, the destination device includes the following modules:

an obtaining module 610, configured to obtain an invocation message, where the invocation message is used to probe the first virtual machine, where the obtaining module 610 is further configured to obtain an identifier of the first virtual machine;
an invocation module, configured to invoke a probe script based on the invocation message, and transfer the identifier of the first virtual machine to the probe script;
a sending module 630, configured to send a probe

packet by running the probe script, where the probe packet is used to probe a status of the first virtual machine in the source device, and the probe packet includes the identifier of the first virtual machine; a receiving module 640, configured to receive feedback information sent by the source device, where the feedback information is used to indicate the status of the first virtual machine; and a determining module 650, configured to determine, based on the feedback information, whether to recreate the first virtual machine in the destination device.

[0176] Optionally, the obtaining module 610, the invocation module 620, the sending module 630, the receiving module 640, and the determining module 650 are configured to perform the operations of the virtual machine fault probe method 200 in this application. For brevity, details are not described herein again.

[0177] FIG. 11 is a schematic block diagram of a source device 700 according to this application. As shown in FIG. 11, the source device includes the following modules:

a receiving module 710, configured to receive a probe packet sent by a destination device, where the probe packet is used to probe a status of the first virtual machine in the source device, and the probe packet includes an identifier of the first virtual machine; a query module 720, configured to query for the status of the first virtual machine based on the probe packet; and a sending module 730, configured to send feedback information to the destination device, where the feedback information is used to indicate the status of the first virtual machine in the source device.

[0178] Optionally, the receiving module 710, the query module 720, and the sending module 730 are configured to perform the operations of the virtual machine fault probe method 200 in this application. For brevity, details are not described herein again.

[0179] The foregoing destination device and source device completely correspond to the destination device and the source device in the method embodiments. Corresponding modules perform corresponding steps. For details, refer to the corresponding method embodiments.

[0180] FIG. 12 is a schematic block diagram of a device 800 according to this application. The device 800 includes:

a memory 810, a processor 820, and an input/output interface 830. The memory 810, the processor 820, and the input/output interface 830 are connected by using an internal connection path. The memory 810 is configured to store a program instruction. The processor 820 is configured to execute the program instruction stored in the memory 810, to control the input/output interface 830 to

receive input data and information, and output data such as an operation result.

[0181] Optionally, when the program instruction is executed, the processor 820 may implement the operations of the method 200. For brevity, details are not described herein again. The input/output interface 830 is configured to perform, under driving of the processor 820, specific signal receiving and sending.

[0182] The device 800 may be the foregoing source device or destination device, and performs operations of the determining module, the query module, and the invocation module. The input/output interface 830 performs corresponding steps of the sending module and the receiving module.

[0183] It should be understood that, in this embodiment of this application, the processor 820 may be a central processing unit (CPU), or the processor 820 may be another general purpose processor, a digital signal processor (DSP), an application-specific integrated circuit (ASIC), a field programmable gate array (FPGA) or another programmable logic device, a discrete gate or a transistor logic device, a discrete hardware component, or the like.

[0184] A person of ordinary skill in the art may be aware that, in combination with units and algorithm steps of the examples described in the embodiments disclosed in this specification, this application may be implemented by electronic hardware or a combination of computer software and electronic hardware. Whether the functions are performed by hardware or software depends on particular applications and design constraints of the technical solutions. A person skilled in the art may use different methods to implement the described functions for each particular application, but it should not be considered that the implementation goes beyond the scope of this application.

[0185] It may be clearly understood by a person skilled in the art that, for the purpose of convenient and brief description, for a detailed working process of the foregoing system, apparatus, and unit, refer to a corresponding process in the foregoing method embodiments, and details are not described herein again.

[0186] In the several embodiments provided in this application, it should be understood that the disclosed system, apparatus, and method may be implemented in other manners. For example, the described apparatus embodiments are merely examples. For example, the unit division is merely logical function division and may be other division during actual implementation. For example, a plurality of units or components may be combined or integrated into another system, or some features may be ignored or not performed. In addition, the displayed or discussed mutual couplings or direct couplings or communication connections may be implemented by using some interfaces. The indirect couplings or communication connections between the apparatuses or units may be implemented in electronic, mechanical, or other forms.

[0187] The units described as separate parts may or

may not be physically separate, and parts displayed as units may or may not be physical units, may be located in one position, or may be distributed on a plurality of network units. Some or all of the units may be selected based on actual requirements to achieve the objectives of the solutions of the embodiments.

[0188] In addition, function units in the embodiments of this application may be integrated into one processing unit, or each of the units may exist alone physically, or two or more units may be integrated into one unit.

[0189] When the functions are implemented in the form of a software function unit and sold or used as an independent product, the functions may be stored in a computer-readable storage medium. Based on such an understanding, the technical solutions of this application essentially, or the part contributing to the prior art, or some of the technical solutions may be implemented in a form of a software product. The computer software product is stored in a storage medium, and includes several instructions for instructing a computer device (which may be a personal computer, a server, a network device, or the like) to perform all or some of the steps of the methods described in the embodiments of this application. The foregoing storage medium includes: any medium that can store program code, such as a USB flash drive, a removable hard disk, a read-only memory (ROM), a random access memory (RAM), a magnetic disk, or an optical disc.

[0190] The foregoing descriptions are merely specific implementations of this application, but are not intended to limit the protection scope of this application. Any variation or replacement readily figured out by a person skilled in the art within the technical scope disclosed in this application shall fall within the protection scope of this application. Therefore, the protection scope of this application shall be subject to the protection scope of the claims.

Claims

1. A virtual machine status probe method in a system comprising

a destination device having a probe port; and
a source device having a probe port and running a first virtual machine, wherein the first virtual machine is located in a first service plane;
wherein the probe port of the source device (130) and the probe port of the destination device (120) are located in a same VLAN;
a management device storing a probe virtual local area network VLAN identifier of the service plane in which the first virtual machine is located;
wherein the method is performed by the destination device (120), and comprises:

when a high availability procedure of the first

virtual machine is triggered,
obtaining (210), (220) from the management device (110) an invocation message comprising an identifier of the first virtual machine and the probe virtual local area network VLAN identifier of the first service plane;

configuring the probe port of the destination device (120) in the first service plane, and configuring the probe virtual VLAN identifier for the probe port of the destination device (120);

sending (240) a probe packet through the probe port of the destination device (120) to the source device (130), wherein the probe packet is used to probe a status of the first virtual machine in the source device (130), the probe packet is a broadcast packet, and the probe packet comprises the obtained identifier of the first virtual machine and the obtained identifier of the probe VLAN;

receiving (280) feedback information sent by the source device (130), wherein the feedback information is used to indicate the status of the first virtual machine; and determining, based on the feedback information, whether to recreate the first virtual machine in the destination device (120).

2. The method according to claim 1, wherein before the obtaining an invocation message, the method further comprises:

sending a first message to a server, wherein the first message is used to request disk storage information of the first virtual machine in the server, and is used to determine a status of the first virtual machine by probing a storage plane of the first virtual machine.

3. The method according to claim 2, the determining, based on the feedback information, whether to recreate the first virtual machine in the destination device (120) comprises:

determining, when the feedback information indicates that the status of the first virtual machine in the source device (130) is normal, not to recreate the first virtual machine in the destination device (120); or
determining, when the feedback information indicates that the status of the first virtual machine in the source device (130) is faulty, to recreate the first virtual machine in the destination device (120).

4. A virtual machine status probe method in a system comprising

a destination device having a tunnel port; and
a source device having a tunnel port and running
a first virtual machine, the method is performed
by a destination device (120), and comprises:

when a high availability procedure of a first
virtual machine is triggered,
obtaining (210), (220), from a management
device (110) an invocation message comprising
an identifier of the first virtual machine;
obtaining, from the management device
based on the identifier of the first virtual machine,
an identifier of the tunnel port corresponding to
the source device;
invoking (230) a probe script based on the
invocation message, and transferring the
identifier of the first virtual machine to the
probe script;
sending (240) a probe packet, by running
the probe script, through the tunnel port of
the destination device (120) to the source
device (130), wherein the probe packet is
used to probe a status of the first virtual machine
in the source device (130), the probe packet
is a unicast packet, and the probe packet
comprises the identifier of the first virtual
machine and the identifier of the tunnel port
corresponding to the source device,
and the source device (130) communicates
with the destination device (120) through an
overlay network;
receiving (280) feedback information sent
by the source device (130), wherein the
feedback information is used to indicate the
status of the first virtual machine; and
determining, based on the feedback information,
whether to recreate the first virtual machine
in the destination device (120).

5. The method according to claim 4, wherein before the
obtaining an invocation message, the method further
comprises:

sending a first message to a server, wherein the first
message is used to request disk storage information
of the first virtual machine in the server, and is used
to determine a status of the first virtual machine by
probing a storage plane of the first virtual machine.

6. The method according to claim 4 or 5, the determining,
based on the feedback information, whether to
recreate the first virtual machine in the destination
device (120) comprises:

determining, when the feedback information indicates
that the status of the first virtual machine in the
source device (130) is normal, not to recreate the
first virtual machine in the destination

device (120); or
determining, when the feedback information indicates
that the status of the first virtual machine in the
source device (130) is faulty, to recreate the first
virtual machine in the destination device (120).

7. A virtual machine status probe method in a system
comprising

a destination device having a probe port; and
a source device having a probe port and running
a first virtual machine, wherein the first virtual
machine is located in a first service plane;
wherein the probe port of the source device
(130) and the probe port of the destination device
(120) are located in a same VLAN;
a management device storing a probe virtual local
area network VLAN identifier of the service
plane in which the first virtual machine is located;
wherein the method is performed by the source
device (130), and comprises:

when a high availability procedure of the first
virtual machine is triggered,
receiving a probe virtual local area network
VLAN identifier that is of a first service plane
in which the first virtual machine is located
and that is sent by a management device
(110);
configuring the probe port in the first service
plane, and configuring the probe VLAN
identifier for the probe port;
receiving (250), through the probe port, a
probe packet sent by a destination device
(120), wherein the probe packet is used to
probe a status of the first virtual machine in
the source device (130), the probe packet
is a broadcast packet, and the probe packet
comprises the identifier of the first virtual
machine and the identifier of the probe
VLAN of the service plane in which the first
virtual machine is located, and the probe
port of the source device (130) and a probe
port of the destination device (120) are located
in a same VLAN;
querying (260) for the status of the first virtual
machine based on the probe packet;
and
sending (270) feedback information to the
destination device (120), wherein the feedback
information is used to indicate the status of
the first virtual machine in the source device
(130), and wherein the destination device
(120) is configured to determine, based on
the feedback information, whether to configure
the first virtual machine in the destination
device (120).

8. A system, wherein the system comprises a destination device (120) and a source device (130), and wherein:

the destination device (120) is configured to perform the virtual machine status probe method according to any one of claims 1 to 3;
the source device (130) is configured to perform the virtual machine status probe method according to claim 7.

9. A system, wherein the system comprises a destination device (120) and a source device (130), and wherein:

the destination device (120) is configured to perform the virtual machine status probe method according to any one of claims 4 to 6;
the source device (130) is configured to perform the virtual machine status probe method according to claim 7.

10. A destination device (120), comprises:

a memory storing at least one program instruction;
a processor, when executing the at least one program instruction in the memory, is configured to perform the virtual machine status probe method according to any one of claims 1 to 3.

11. A destination device (120), comprises:

a memory storing at least one program instruction;
a processor, when executing the at least one program instruction in the memory, is configured to perform the virtual machine status probe method according to any one of claims 4 to 6.

12. A source device (130), comprises:

a memory storing at least one program instruction;
a processor, when executing the at least one program instruction in the memory, is configured to perform the virtual machine status probe method according to any one of claims 7.

13. A storage medium, wherein the storage medium stores at least one instruction, and the at least one instruction is loaded and executed by a processor to implement the virtual machine status probe method according to any one of claims 1 to 3.

14. A storage medium, wherein the storage medium stores at least one instruction, and the at least one instruction is loaded and executed by a processor to

implement the virtual machine status probe method according to any one of claims 4 to 6.

15. A storage medium, wherein the storage medium stores at least one instruction, and the at least one instruction is loaded and executed by a processor to implement the virtual machine status probe method according to any one of claims 7.

Patentansprüche

1. Verfahren zur Statusprüfung einer virtuellen Maschine in einem System, umfassend

eine Zielvorrichtung, die einen Prüfport aufweist; und

eine Quellvorrichtung, die einen Prüfport aufweist und auf der eine erste virtuelle Maschine ausgeführt wird, wobei sich die erste virtuelle Maschine in einer ersten Dienstebene befindet; wobei sich der Prüfport der Quellvorrichtung (130) und der Prüfport der Zielvorrichtung (120) in demselben VLAN befinden;

eine Verwaltungsvorrichtung, die eine VLAN-Prüfkennung eines virtuellen lokalen Netzwerks der Dienstebene speichert, in der sich die erste virtuelle Maschine befindet;

wobei das Verfahren durch die Zielvorrichtung (120) durchgeführt wird und Folgendes umfasst:

wenn eine Hochverfügbarkeitsprozedur der ersten virtuellen Maschine ausgelöst wird, Erlangen (210), (220) einer Aufrufnachricht, die eine Kennung der ersten virtuellen Maschine und die VLAN-Prüfkennung des virtuellen lokalen Netzwerks der ersten Dienstebene umfasst, von der Verwaltungsvorrichtung (110);

Konfigurieren des Prüfports der Zielvorrichtung (120) in der ersten Dienstebene und Konfigurieren der virtuellen VLAN-Prüfkennung für den Prüfport der Zielvorrichtung (120);

Senden (240) eines Prüfpakets über den Prüfport der Zielvorrichtung (120) an die Quellvorrichtung (130), wobei das Prüfpaket verwendet wird, um einen Status der ersten virtuellen Maschine in der Quellvorrichtung (130) zu prüfen, das Prüfpaket ein Broadcast-Paket ist und das Prüfpaket die erlangte Kennung der ersten virtuellen Maschine und die erlangte Kennung des Prüf-VLAN umfasst;

Empfangen (280) von Rückmeldungsinformationen, die von der Quellvorrichtung (130) gesendet werden, wobei die Rückmeldungsinformationen verwendet wer-

- den, um den Status der ersten virtuellen Maschine anzugeben; und
Bestimmen, ob die erste virtuelle Maschine in der Zielvorrichtung (120) neu erstellt werden soll, basierend auf den Rückmeldungs-
informationen. 5
2. Verfahren nach Anspruch 1, wobei das Verfahren vor dem Erlangen einer Aufrufnachricht ferner Folgendes umfasst: 10
Senden einer ersten Nachricht an einen Server, wobei die erste Nachricht verwendet wird, um Festplattenspeicherinformationen der ersten virtuellen Maschine im Server anzufordern, und verwendet wird, um einen Status der ersten virtuellen Maschine durch Prüfen einer Speicherebene der ersten virtuellen Maschine zu bestimmen. 15
3. Verfahren nach Anspruch 2, wobei das Bestimmen, ob die erste virtuelle Maschine in der Zielvorrichtung (120) neu erstellt werden soll, basierend auf den Rückmeldungs-
informationen, Folgendes umfasst: 20
Bestimmen, wenn die Rückmeldungs-
informationen angeben, dass der Status der ersten virtuellen Maschine in der Quellvorrichtung (130) normal ist, dass die erste virtuelle Maschine in der Zielvorrichtung (120) nicht neu erstellt werden soll; oder 25
Bestimmen, wenn die Rückmeldungs-
informationen angeben, dass der Status der ersten virtuellen Maschine in der Quellvorrichtung (130) fehlerhaft ist, dass die erste virtuelle Maschine in der Zielvorrichtung (120) neu erstellt werden soll. 30 35
4. Verfahren zur Statusprüfung einer virtuellen Maschine in einem System, umfassend
eine Zielvorrichtung, die einen Tunnelport aufweist; und 40
eine Quellvorrichtung, die einen Tunnelport aufweist und auf der eine erste virtuelle Maschine ausgeführt wird, wobei das Verfahren durch eine Zielvorrichtung (120) durchgeführt wird und Folgendes umfasst: 45
wenn eine Hochverfügbarkeitsprozedur einer ersten virtuellen Maschine ausgelöst wird, 50
Erlangen (210), (220) einer Aufrufnachricht, die eine Kennung der ersten virtuellen Maschine umfasst, von einer Verwaltungsvorrichtung (110);
Erlangen einer Kennung des Tunnelports, der der Quellvorrichtung entspricht, von der Verwaltungsvorrichtung basierend auf der Kennung der ersten virtuellen Maschine; 55
- Aufrufen (230) eines Prüfskripts basierend auf der Aufrufnachricht und Übertragen der Kennung der ersten virtuellen Maschine an das Prüfskript;
Senden (240) eines Prüfpakets durch Ausführen des Prüfskripts über den Tunnelport der Zielvorrichtung (120) an die Quellvorrichtung (130), wobei das Prüfpaket zum Prüfen eines Status der ersten virtuellen Maschine in der Quellvorrichtung (130) verwendet wird, das Prüfpaket ein Unicast-Paket ist und das Prüfpaket die Kennung der ersten virtuellen Maschine und die Kennung des der Quellvorrichtung entsprechenden Tunnelports umfasst und die Quellvorrichtung (130) über ein Overlay-Netzwerk mit der Zielvorrichtung (120) kommuniziert;
Empfangen (280) von Rückmeldungs-
informationen, die von der Quellvorrichtung (130) gesendet werden, wobei die Rückmeldungs-
informationen verwendet werden, um den Status der ersten virtuellen Maschine anzugeben; und
Bestimmen, ob die erste virtuelle Maschine in der Zielvorrichtung (120) neu erstellt werden soll, basierend auf den Rückmeldungs-
informationen.
5. Verfahren nach Anspruch 4, wobei das Verfahren vor dem Erlangen einer Aufrufnachricht ferner Folgendes umfasst:
Senden einer ersten Nachricht an einen Server, wobei die erste Nachricht verwendet wird, um Festplattenspeicherinformationen der ersten virtuellen Maschine im Server anzufordern, und verwendet wird, um einen Status der ersten virtuellen Maschine durch Prüfen einer Speicherebene der ersten virtuellen Maschine zu bestimmen.
6. Verfahren nach Anspruch 4 oder 5, wobei das Bestimmen, ob die erste virtuelle Maschine in der Zielvorrichtung (120) neu erstellt werden soll, basierend auf den Rückmeldungs-
informationen, Folgendes umfasst:
Bestimmen, wenn die Rückmeldungs-
informationen angeben, dass der Status der ersten virtuellen Maschine in der Quellvorrichtung (130) normal ist, dass die erste virtuelle Maschine in der Zielvorrichtung (120) nicht neu erstellt werden soll; oder
Bestimmen, wenn die Rückmeldungs-
informationen angeben, dass der Status der ersten virtuellen Maschine in der Quellvorrichtung (130) fehlerhaft ist, dass die erste virtuelle Maschine in der Zielvorrichtung (120) neu erstellt werden soll.

7. Verfahren zur Statusprüfung einer virtuellen Maschine in einem System, umfassend

eine Zielvorrichtung, die einen Prüfport aufweist; und
 eine Quellvorrichtung, die einen Prüfport aufweist und auf der eine erste virtuelle Maschine ausgeführt wird, wobei sich die erste virtuelle Maschine in einer ersten Dienstebene befindet; wobei sich der Prüfport der Quellvorrichtung (130) und der Prüfport der Zielvorrichtung (120) in demselben VLAN befinden;
 eine Verwaltungsvorrichtung, die eine VLAN-Prüfkennung eines virtuellen lokalen Netzwerks der Dienstebene speichert, in der sich die erste virtuelle Maschine befindet;
 wobei das Verfahren durch die Quellvorrichtung (130) durchgeführt wird und Folgendes umfasst:

wenn eine Hochverfügbarkeitsprozedur der ersten virtuellen Maschine ausgelöst wird, Empfangen einer VLAN-Prüfkennung eines virtuellen lokalen Netzwerks, die zu einer ersten Dienstebene gehört, in der sich die erste virtuelle Maschine befindet, und die durch eine Verwaltungsvorrichtung (110) gesendet wird;

Konfigurieren des Prüfports in der ersten Dienstebene und Konfigurieren der VLAN-Prüfkennung für den Prüfport;

Empfangen (250) eines durch eine Zielvorrichtung (120) gesendeten Prüfpakets über den Prüfport, wobei das Prüfpaket verwendet wird, um einen Status der ersten virtuellen Maschine in der Quellvorrichtung (130) zu prüfen, das Prüfpaket ein Broadcast-Paket ist und das Prüfpaket die Kennung der ersten virtuellen Maschine und die Kennung des Prüf-VLAN der Dienstebene umfasst, in der sich die erste virtuelle Maschine befindet, und sich der Prüfport der Quellvorrichtung (130) und ein Prüfport der Zielvorrichtung (120) in demselben VLAN befinden;

Abfragen (260) des Status der ersten virtuellen Maschine basierend auf dem Prüfpaket; und

Senden (270) von Rückmeldungsinformationen an die Zielvorrichtung (120), wobei die Rückmeldungsinformationen verwendet werden, um den Status der ersten virtuellen Maschine in der Quellvorrichtung (130) anzugeben, und wobei die Zielvorrichtung (120) dazu konfiguriert ist, basierend auf den Rückmeldungsinformationen zu bestimmen, ob die erste virtuelle Maschine in der Zielvorrichtung (120) konfiguriert werden soll.

8. System, wobei das System eine Zielvorrichtung (120) und eine Quellvorrichtung (130) umfasst und wobei:

die Zielvorrichtung (120) dazu konfiguriert ist, das Verfahren zur Statusprüfung einer virtuellen Maschine nach einem der Ansprüche 1 bis 3 durchzuführen;

die Quellvorrichtung (130) dazu konfiguriert ist, das Verfahren zur Statusprüfung einer virtuellen Maschine nach Anspruch 7 durchzuführen.

9. System, wobei das System eine Zielvorrichtung (120) und eine Quellvorrichtung (130) umfasst und wobei:

die Zielvorrichtung (120) dazu konfiguriert ist, das Verfahren zur Statusprüfung einer virtuellen Maschine nach einem der Ansprüche 4 bis 6 durchzuführen;

die Quellvorrichtung (130) dazu konfiguriert ist, das Verfahren zur Statusprüfung einer virtuellen Maschine nach Anspruch 7 durchzuführen.

10. Zielvorrichtung (120), umfassend:

einen Speicher, in dem mindestens eine Programmanweisung gespeichert ist;

einen Prozessor, der beim Ausführen der mindestens einen Programmanweisung im Speicher dazu konfiguriert ist, das Verfahren zur Statusprüfung einer virtuellen Maschine nach einem der Ansprüche 1 bis 3 durchzuführen.

11. Zielvorrichtung (120), umfassend:

einen Speicher, in dem mindestens eine Programmanweisung gespeichert ist;

einen Prozessor, der beim Ausführen der mindestens einen Programmanweisung im Speicher dazu konfiguriert ist, das Verfahren zur Statusprüfung einer virtuellen Maschine nach einem der Ansprüche 4 bis 6 durchzuführen.

12. Quellvorrichtung (130), umfassend:

einen Speicher, in dem mindestens eine Programmanweisung gespeichert ist;

einen Prozessor, der beim Ausführen der mindestens einen Programmanweisung im Speicher dazu konfiguriert ist, das Verfahren zur Statusprüfung einer virtuellen Maschine nach Anspruch 7 durchzuführen.

13. Speichermedium, wobei in dem Speichermedium mindestens eine Anweisung gespeichert ist, und die mindestens eine Anweisung durch einen Prozessor geladen und ausgeführt wird, um das Verfahren zur

Statusprüfung einer virtuellen Maschine nach einem der Ansprüche 1 bis 3 zu implementieren.

14. Speichermedium, wobei in dem Speichermedium mindestens eine Anweisung gespeichert ist, und die mindestens eine Anweisung durch einen Prozessor geladen und ausgeführt wird, um das Verfahren zur Statusprüfung einer virtuellen Maschine nach einem der Ansprüche 4 bis 6 zu implementieren. 5
15. Speichermedium, wobei in dem Speichermedium mindestens eine Anweisung gespeichert ist, und die mindestens eine Anweisung durch einen Prozessor geladen und ausgeführt wird, um das Verfahren zur Statusprüfung einer virtuellen Maschine nach Anspruch 7 zu implementieren. 10 15

Revendications

1. Procédé de vérification d'état de machine virtuelle dans un système comprenant 20

un dispositif de destination ayant un port de sonde ; et 25

un dispositif source ayant un port de sonde et exécutant une première machine virtuelle, dans lequel la première machine virtuelle est située dans un premier plan de service ; dans lequel le port de sonde du dispositif source (130) et le port de sonde du dispositif de destination (120) sont situés dans un même VLAN ; un dispositif de gestion stockant un identifiant de réseau local virtuel, VLAN, de sonde du plan de service dans lequel se trouve la première machine virtuelle ; 30 35

dans lequel le procédé est exécuté par le dispositif de destination (120) et comprend :

lorsqu'une procédure de haute disponibilité de la première machine virtuelle est déclenchée, 40

l'obtention (210), (220) du dispositif de gestion (110) d'un message d'invocation comprenant un identifiant de la première machine virtuelle et l'identifiant de réseau local virtuel, VLAN, de sonde du premier plan de service ; 45

la configuration du port de sonde du dispositif de destination (120) dans le premier plan de service, et la configuration de l'identifiant VLAN virtuel de sonde pour le port de sonde du dispositif de destination (120) ; 50

l'envoi (240) d'un paquet de sonde via le port de sonde du dispositif de destination (120) au dispositif source (130), dans lequel le paquet de sonde est utilisé pour sonder un état de la première machine virtuelle 55

dans le dispositif source (130), le paquet de sonde est un paquet de diffusion, et le paquet de sonde comprend l'identifiant obtenu de la première machine virtuelle et l'identifiant obtenu du VLAN de sonde ;

la réception (280) des informations de retour envoyées par le dispositif source (130), les informations de retour étant utilisées pour indiquer l'état de la première machine virtuelle ; et

la détermination, sur la base des informations de retour, s'il convient de recréer la première machine virtuelle dans le dispositif de destination (120).

2. Procédé selon la revendication 1, dans lequel avant l'obtention d'un message d'invocation, le procédé comprend également :

l'envoi d'un premier message à un serveur, dans lequel le premier message est utilisé pour demander des informations de stockage sur disque de la première machine virtuelle dans le serveur, et est utilisé pour déterminer un état de la première machine virtuelle en sondant un plan de stockage de la première machine virtuelle.

3. Procédé selon la revendication 2, dans lequel la détermination, sur la base des informations de retour, du fait de savoir s'il convient de recréer la première machine virtuelle dans le dispositif de destination (120) comprend :

la détermination, lorsque les informations de retour indiquent que l'état de la première machine virtuelle dans le dispositif source (130) est normal, du fait de ne pas recréer la première machine virtuelle dans le dispositif de destination (120) ; ou

la détermination, lorsque les informations de retour indiquent que l'état de la première machine virtuelle dans le dispositif source (130) est défectueux, du fait de recréer la première machine virtuelle dans le dispositif de destination (120).

4. Procédé de vérification d'état de machine virtuelle dans un système comprenant

un dispositif de destination ayant un port tunnel ; et

un dispositif source ayant un port tunnel et exécutant une première machine virtuelle, le procédé est exécuté par un dispositif de destination (120) et comprend :

lorsqu'une procédure de haute disponibilité d'une première machine virtuelle est déclenchée, l'obtention (210), (220), auprès d'un dispo-

- sitif de gestion (110), d'un message d'invocation comprenant un identifiant de la première machine virtuelle ;
 l'obtention, auprès du dispositif de gestion sur la base de l'identifiant de la première machine virtuelle, d'un identifiant du port tunnel correspondant au dispositif source ;
 l'appel (230) d'un script de sonde sur la base du message d'invocation, et le transfert de l'identifiant de la première machine virtuelle au script de sonde ;
 l'envoi (240) d'un paquet de sonde, en exécutant le script de sonde, via le port tunnel du dispositif de destination (120) vers le dispositif source (130), dans lequel le paquet de sonde est utilisé pour sonder un état de la première machine virtuelle dans le dispositif source (130), le paquet de sonde est un paquet unicast, et le paquet de sonde comprend l'identifiant de la première machine virtuelle et l'identifiant du port tunnel correspondant au dispositif source, et le dispositif source (130) communique avec le dispositif de destination (120) via un réseau superposé ;
 la réception (280) des informations de retour envoyées par le dispositif source (130), dans lequel les informations de retour sont utilisées pour indiquer l'état de la première machine virtuelle ; et
 la détermination, sur la base des informations de retour, du fait de savoir s'il convient de recréer la première machine virtuelle dans le dispositif de destination (120).
5. Procédé selon la revendication 4, dans lequel avant l'obtention d'un message d'invocation, le procédé comprend également :
 l'envoi d'un premier message à un serveur, dans lequel le premier message est utilisé pour demander des informations de stockage sur disque de la première machine virtuelle dans le serveur, et est utilisé pour déterminer un état de la première machine virtuelle en sondant un plan de stockage de la première machine virtuelle.
6. Procédé selon la revendication 4 ou 5, dans lequel la détermination, sur la base des informations de retour, du fait de savoir s'il convient de recréer la première machine virtuelle dans le dispositif de destination (120) comprend :
- la détermination, lorsque les informations de retour indiquent que l'état de la première machine virtuelle dans le dispositif source (130) est normal, du fait de ne pas recréer la première machine virtuelle dans le dispositif de destination (120) ; ou

la détermination, lorsque les informations de retour indiquent que l'état de la première machine virtuelle dans le dispositif source (130) est défectueux, du fait de recréer la première machine virtuelle dans le dispositif de destination (120).

7. Procédé de vérification d'état de machine virtuelle dans un système comprenant

un dispositif de destination ayant un port de sonde ; et
 un dispositif source ayant un port de sonde et exécutant une première machine virtuelle, dans lequel la première machine virtuelle est située dans un premier plan de service ;
 dans lequel le port de sonde du dispositif source (130) et le port de sonde du dispositif de destination (120) sont situés dans un même VLAN ;
 un dispositif de gestion stockant un identifiant de réseau local virtuel, VLAN, de sonde du plan de service dans lequel se trouve la première machine virtuelle ;
 dans lequel le procédé est exécuté par le dispositif source (130) et comprend :

lorsqu'une procédure de haute disponibilité de la première machine virtuelle est déclenchée,

la réception d'un identifiant de réseau local virtuel, VLAN, de sonde qui appartient à un premier plan de service dans lequel se trouve la première machine virtuelle et qui est envoyé par un dispositif de gestion (110) ;
 la configuration du port de sonde dans le premier plan de service, et la configuration de l'identifiant VLAN de sonde pour le port de sonde ;

la réception (250), via le port de sonde, d'un paquet de sonde envoyé par un dispositif de destination (120), dans lequel le paquet de sonde est utilisé pour sonder un état de la première machine virtuelle dans le dispositif source (130), le paquet de sonde est un paquet de diffusion, et le paquet de sonde comprend l'identifiant de la première machine virtuelle et l'identifiant VLAN de sonde du plan de service dans lequel se trouve la première machine virtuelle, et le port de sonde du dispositif source (130) et un port de sonde du dispositif de destination (120) sont situés dans un même VLAN ;
 l'interrogation (260) de l'état de la première machine virtuelle sur la base du paquet de sonde ; et

l'envoi (270) des informations de retour au dispositif de destination (120), dans lequel les informations de retour sont utilisées pour indiquer l'état de la première machine

- virtuelle dans le dispositif source (130), et dans lequel le dispositif de destination (120) est configuré pour déterminer, sur la base des informations de retour, s'il faut configurer la première machine virtuelle dans le dispositif de destination (120). 5
8. Système, dans lequel le système comprend un dispositif de destination (120) et un dispositif source (130), et dans lequel : 10
- le dispositif de destination (120) est configuré pour exécuter le procédé de vérification d'état de machine virtuelle selon l'une quelconque des revendications 1 à 3 ; 15
- le dispositif source (130) est configuré pour exécuter le procédé de vérification d'état de machine virtuelle selon la revendication 7.
9. Système, dans lequel le système comprend un dispositif de destination (120) et un dispositif source (130), et dans lequel : 20
- le dispositif de destination (120) est configuré pour exécuter le procédé de vérification d'état de machine virtuelle selon l'une quelconque des revendications 4 à 6 ; 25
- le dispositif source (130) est configuré pour exécuter le procédé de vérification d'état de machine virtuelle selon la revendication 7. 30
10. Dispositif de destination (120), comprenant :
- une mémoire stockant au moins une instruction de programme ; 35
- un processeur, lors de l'exécution de l'au moins une instruction de programme dans la mémoire, est configuré pour exécuter le procédé de vérification d'état de machine virtuelle selon l'une quelconque des revendications 1 à 3. 40
11. Dispositif de destination (120), comprenant :
- une mémoire stockant au moins une instruction de programme ; 45
- un processeur, lors de l'exécution de l'au moins une instruction de programme dans la mémoire, est configuré pour exécuter le procédé de vérification d'état de machine virtuelle selon l'une quelconque des revendications 4 à 6. 50
12. Dispositif source (130) comprenant :
- une mémoire stockant au moins une instruction de programme ; 55
- un processeur, lors de l'exécution de l'au moins une instruction de programme dans la mémoire, est configuré pour exécuter le procédé de vérification d'état de machine virtuelle selon la revendication 7.
13. Support de stockage, dans lequel le support de stockage stocke au moins une instruction, et l'au moins une instruction est chargée et exécutée par un processeur pour mettre en oeuvre le procédé de vérification d'état de machine virtuelle selon l'une quelconque des revendications 1 à 3.
14. Support de stockage, dans lequel le support de stockage stocke au moins une instruction, et l'au moins une instruction est chargée et exécutée par un processeur pour mettre en oeuvre le procédé de vérification d'état de machine virtuelle selon l'une quelconque des revendications 4 à 6.
15. Support de stockage, dans lequel le support de stockage stocke au moins une instruction, et l'au moins une instruction est chargée et exécutée par un processeur pour mettre en oeuvre le procédé de vérification d'état de machine virtuelle selon la revendication 7.

100

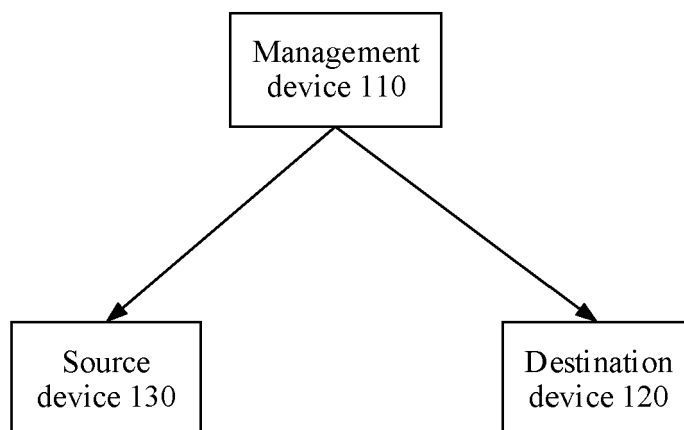


FIG. 1

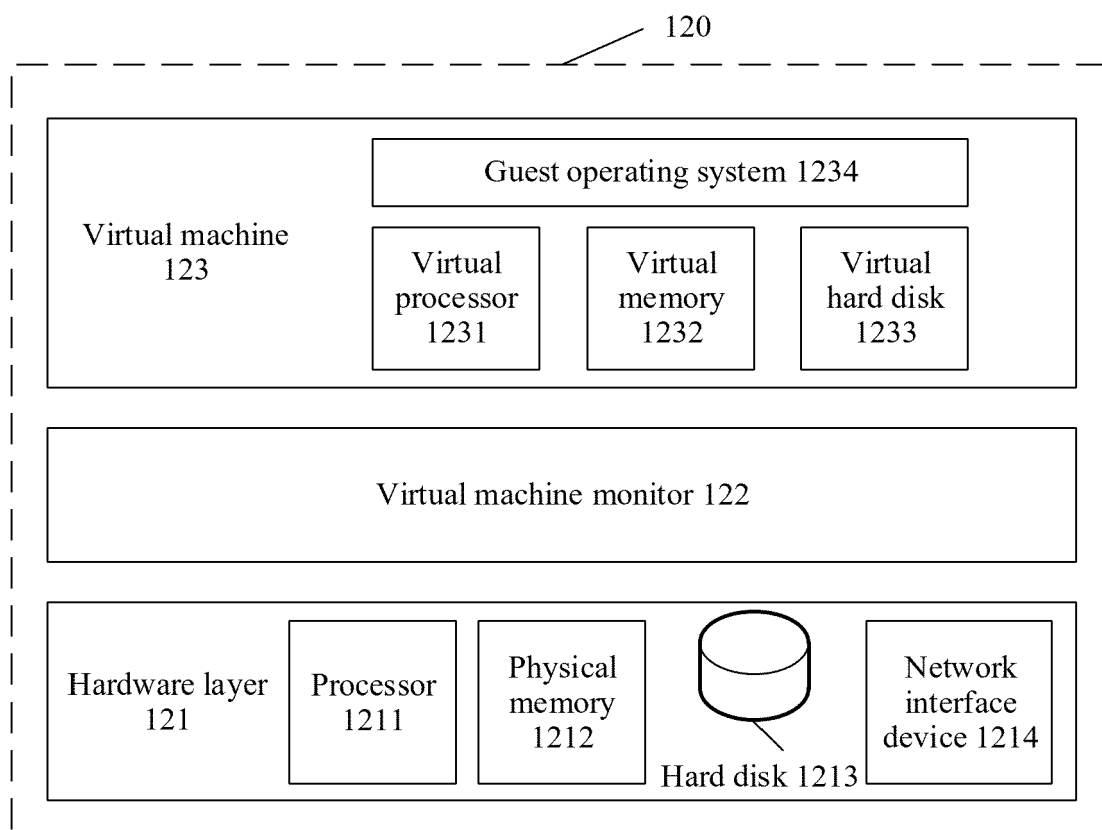


FIG. 2

200

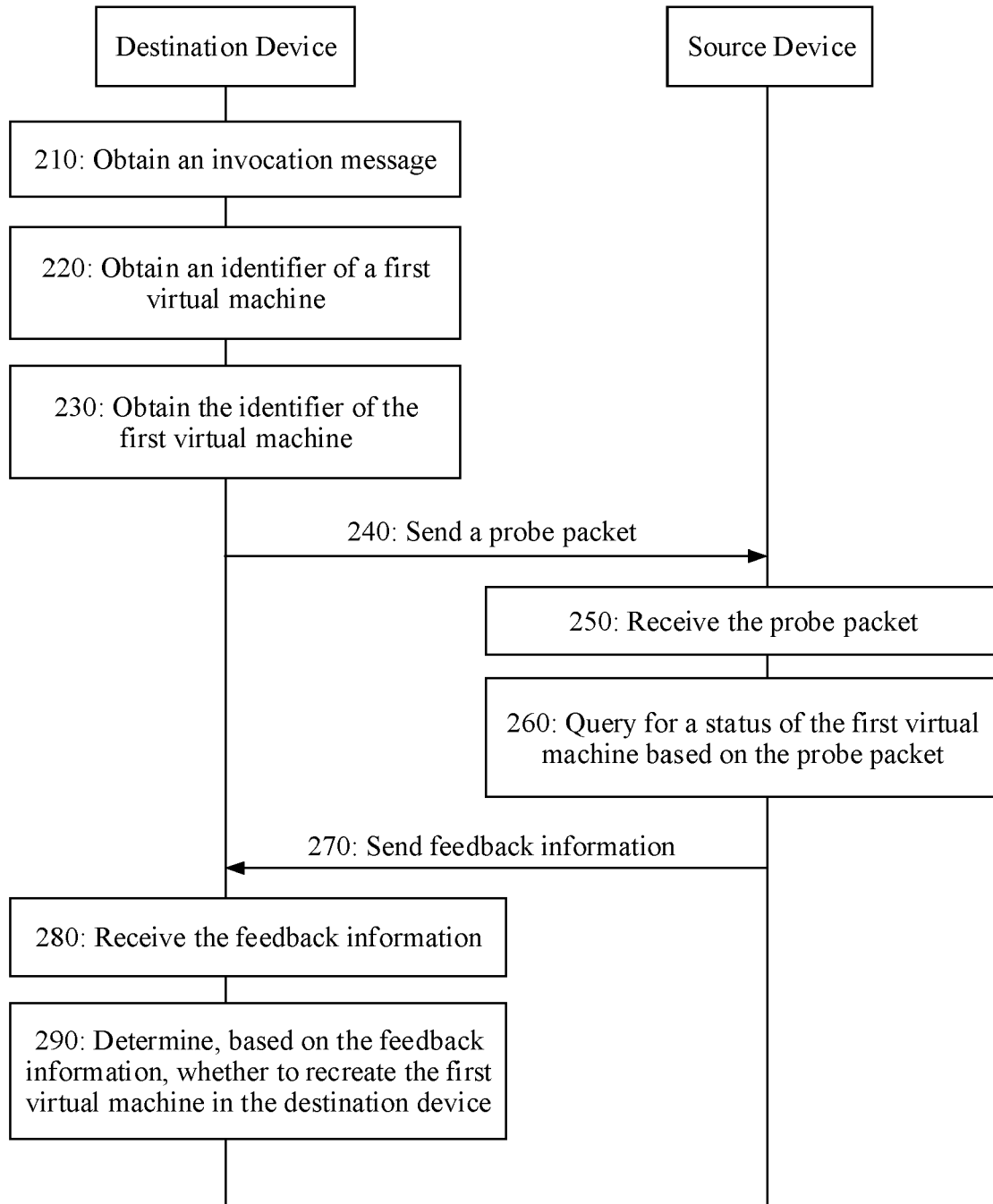


FIG. 3

300

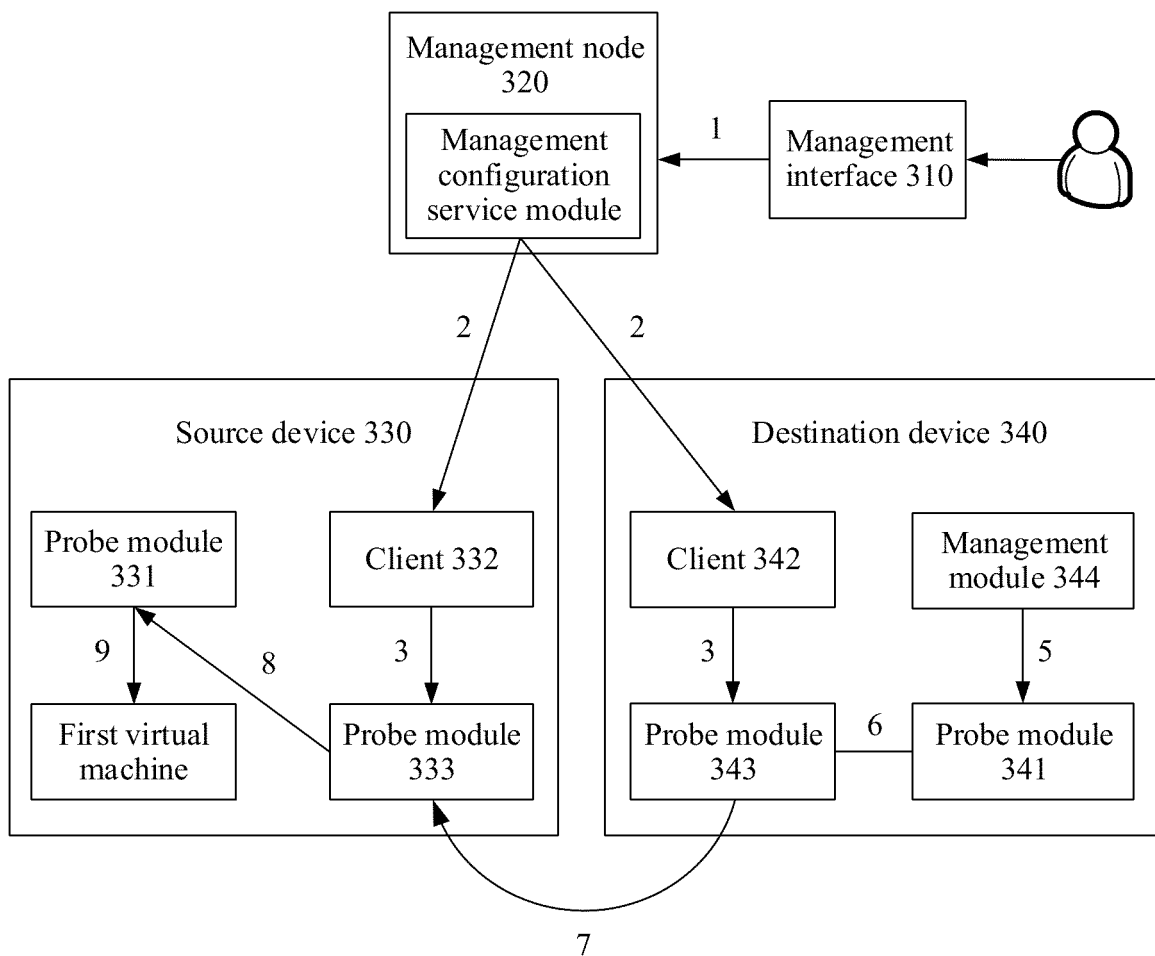


FIG. 4

300

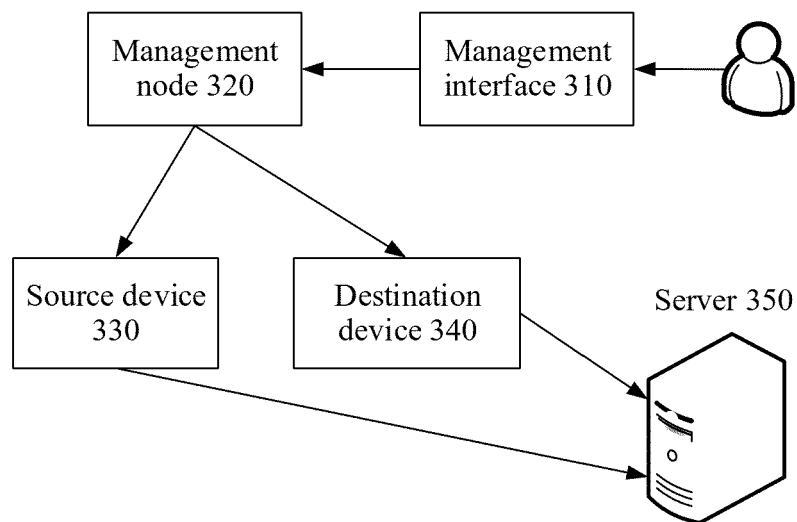


FIG. 5

400

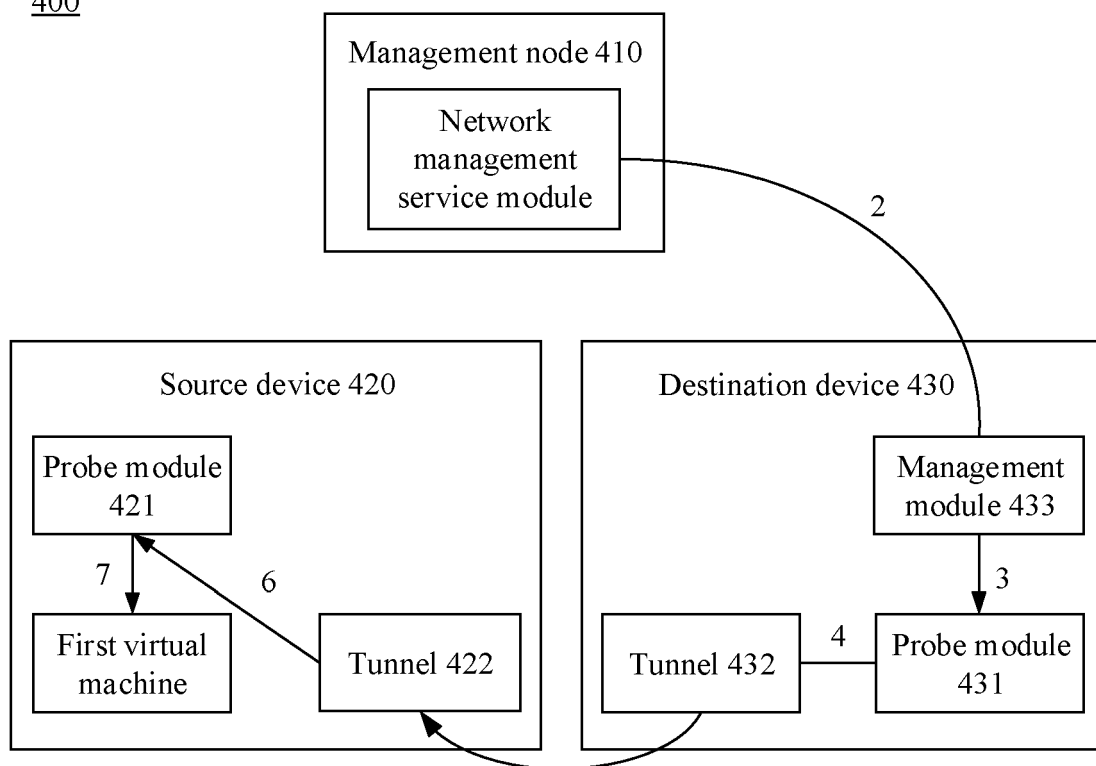


FIG. 6

500

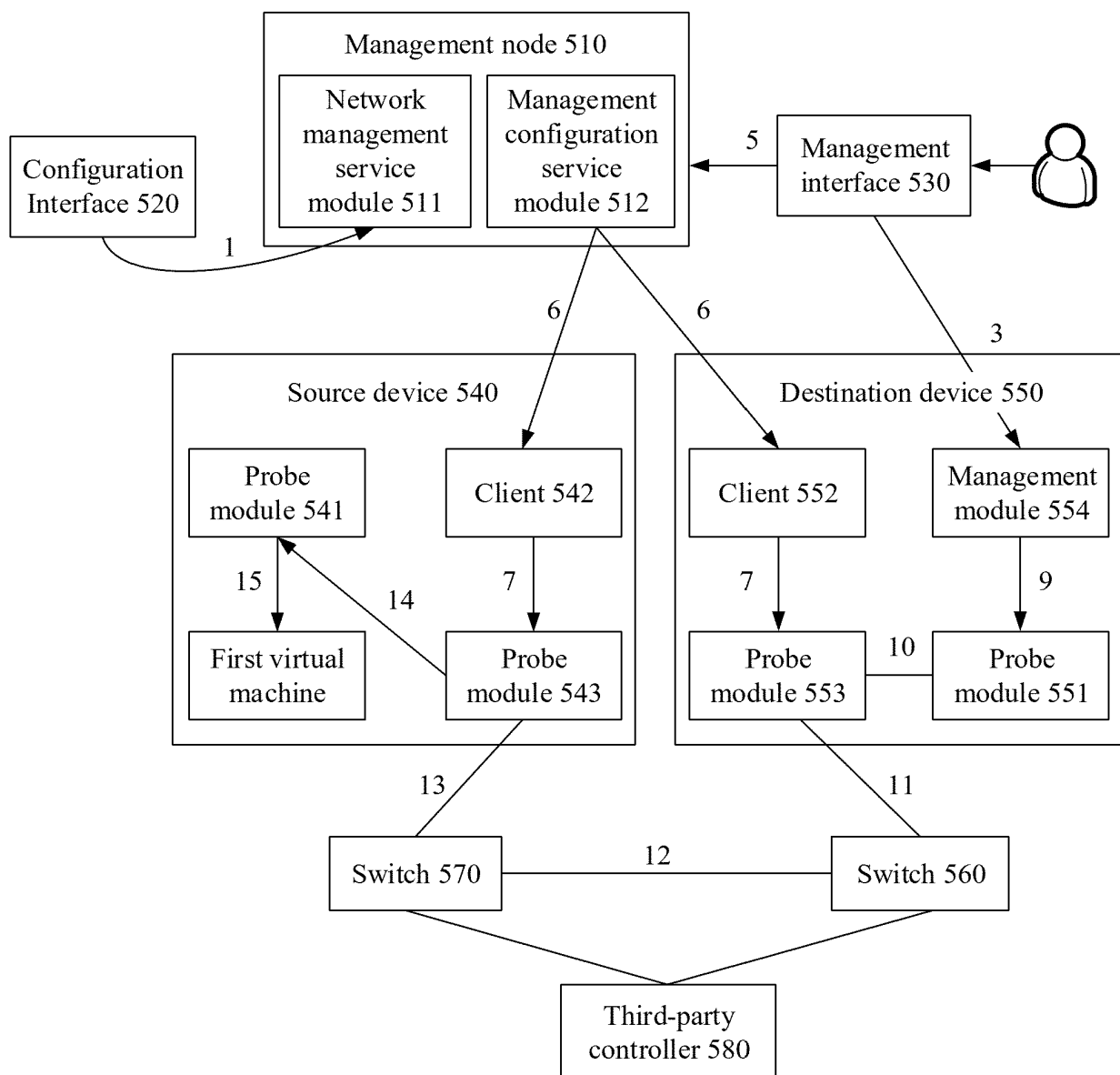


FIG. 7

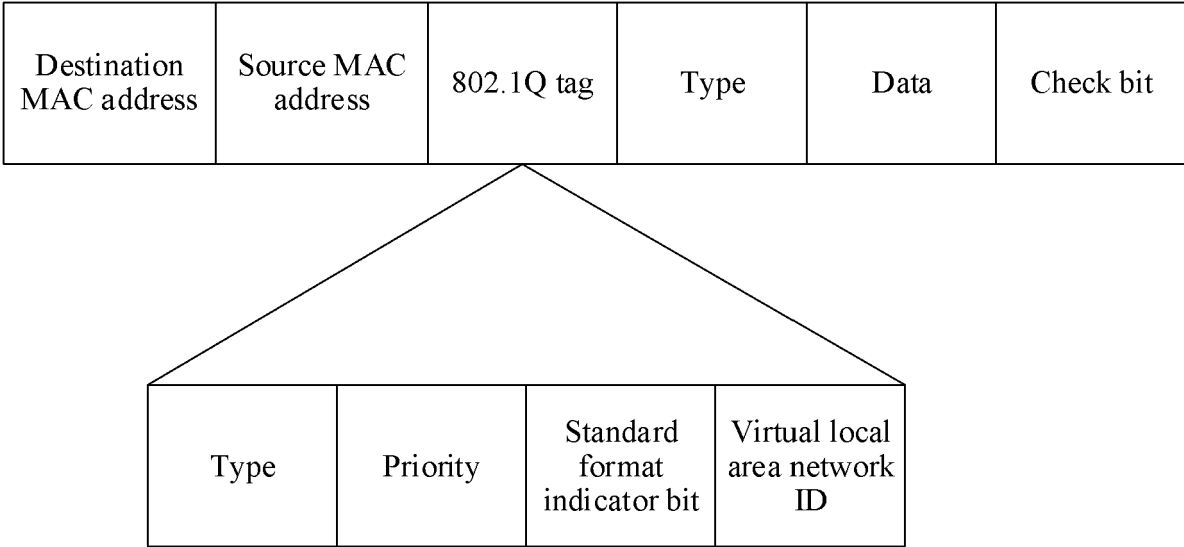


FIG. 8

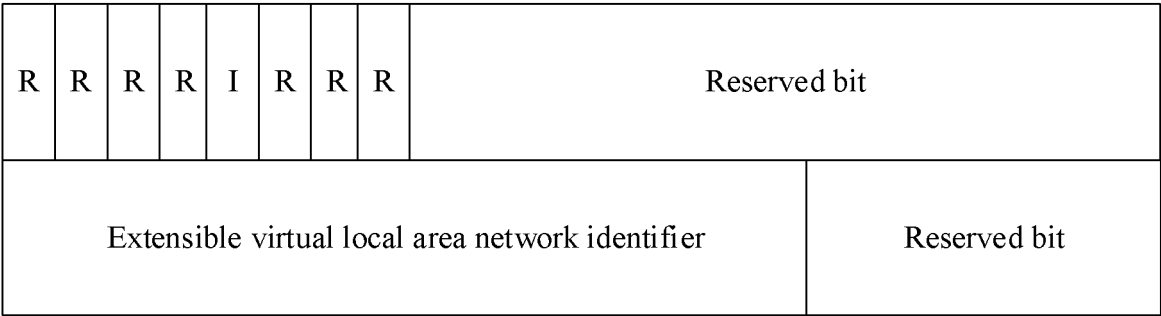


FIG. 9

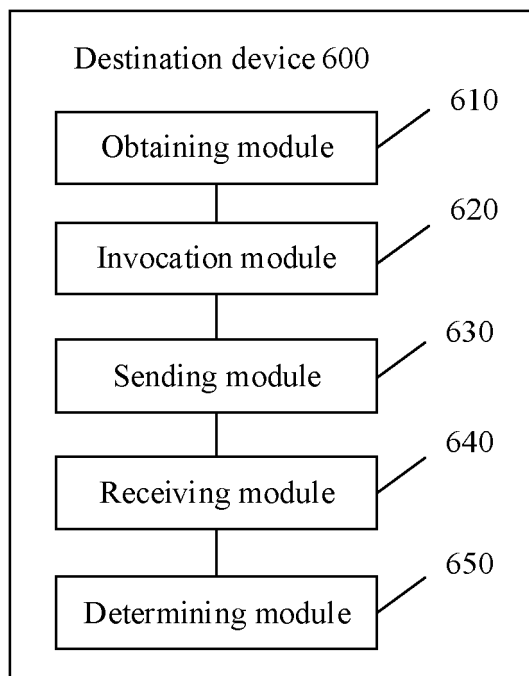


FIG. 10

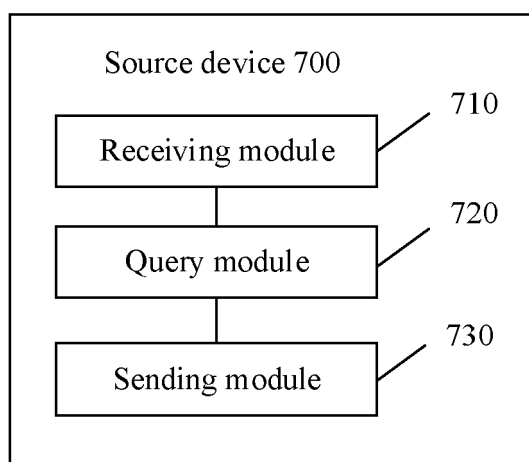


FIG. 11

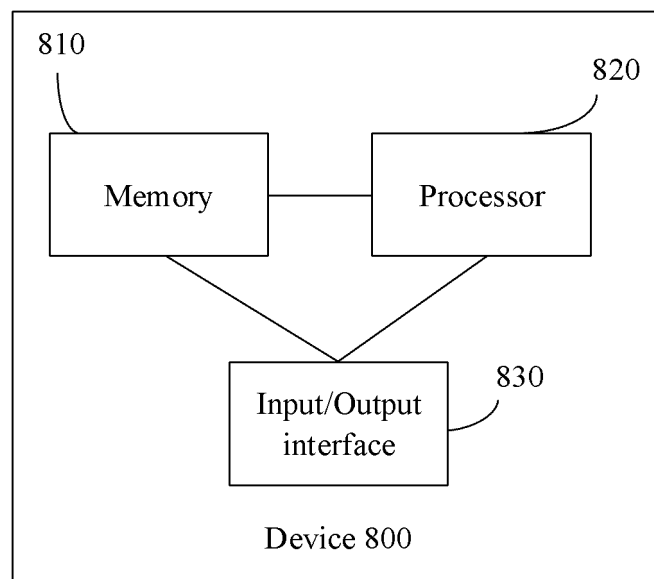


FIG. 12

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 2017168903 A1 [0004]
- US 2017257266 A1 [0005]